
INPUT
Publicación práctica para usuarios de
sinclair
VERANO 87 Precio 375 Ptas
Año 2 Número 21
MAPA DE SPIRITS
TODO SOBRE RANARAMA
ROMPECABEZAS DE CUBOS Y NUMEROS
GRAFICOS ESTADISTICOS
COMO DOMINAR LOS BUCLES
TRINITRON
SONY
sinclair
128K
ZX Spectrum +2
DATACORDER



AÑO 2 NUMERO 21

DIRECTOR: Manuel Pérez
DIRECTOR DE ARTE: Luis F. Balaguer
REALIZACION GRAFICA: Didac Tudela
COLABORADORES: José Vila, Antonio Pliego, Xavier Ferrer, Ernesto del Valle, Equipo Molisoft, Ramón Rabaso, Antonio Taratiel, Jaime Mardones, Carles Bartolomé, Angels Alvarez
FOTOGRAFIA: Ernesto Walfisch, Joan Boada

INPUT Sinclair es una publicación de PLANETA-DE AGOSTINI, S.A.
GERENTE DIVISION DE REVISTAS: Sebastián Martínez

PUBLICIDAD: José Real-Grupo Jota
Madrid: c./ General Varela, 35
Teléf. 270 47 02/03
Barcelona: Avda. de Sarriá, 11-13, 1.°
Teléf. 250 23 99

FOTOMECANICA: TECFA, S.A.

IMPRESION: Sirven Grafic
c./ Gran Vía, 754-756, 08013 Barcelona
Depósito legal: B. 38.115-1986

SUSCRIPCIONES: EDISA
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12

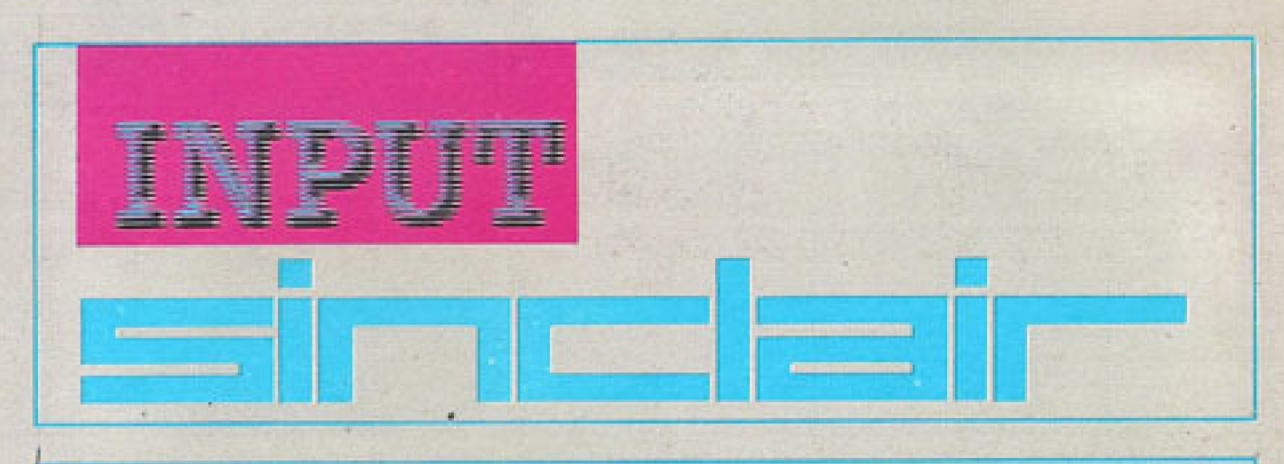
REDACCION:
Aribau, 185, 1.°
08021 Barcelona

DISTRIBUIDORA:
R.B.A. PROMOTORA DE EDICIONES, S.A.
Calle B, n.° 11. Sector B, Zona Franca
08004 Barcelona

El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

INPUT Sinclair es una publicación controlada por





SUMARIO

EDITORIAL

VACACIONES

Se acercan las vacaciones de verano y las noches estivales se prevén largas, cálidas y fantasiosas. Los ordenadores serán el centro de atención de todas las noctámbulas miradas, convirtiendo los aburridos mitines/veladas en fiestas multicolores de sonidos, música y explosiones.

Poco a poco, los joysticks ocupan sus puertos/puestos, el cansancio se va adueñando de nuestras muñecas y los reflejos decaen al paso del tiempo. Llega el final y las despedidas, regresando el silencio a la paz del hogar, a la pantalla del monitor.

El ordenador, compañero de desventuras alienígenas, descansa de sus batallas galácticas interestelares. Las luces se apagan y una nueva etapa comienza, más intimista, más satisfactoria y provechosa: llegar al corazón del microprocesador, seducir al programa residente en memoria, descubrir sus interioridades... Eso es lo que podríamos llamar seducción de un programa.

La noche es un buen estadio para conseguir records, olímpicos o mentales («mens sana in corpore sano»), un circuito donde correr contra reloj contra otros programadores, un «campus» de soledad y duelo entre la máquina y el hombre.

En esas horas, el BASIC se convierte en delator y el código máquina en enemigo. Las sentencias chocan contra los errores sintácticos y los desensambladores son paladines de causas aun por defender. La lucha es ardua, cruel, salvaje,... los párpados sucumben al sueño, la mente obnubilada es presa de ciegas fantasías y lo que creíamos poder hacer se convierte en utopía...

Una nueva mañana clarea en el horizonte y el RETURN del cansancio nos RESETEA hacia otra SUBRUTINA: la del sueño.

AMPLIA TU CURSO DE MECANOGRAFIA

- PARA ADQUIRIR UN RITMO REGULAR
- INCREMENTO DE LA VELOCIDAD Y DE LA PRECISION
- UN CONSEJO IMPORTANTE

Esta vez puedes extender tu pericia a las teclas de números y a todos los restantes símbolos del teclado. Hay, además, un programa de juego que te ayudará a aumentar todavía más tu rapidez en la escritura.

Si has trabajado ya con el instructor mecanográfico que publicamos en INPUT, estarás familiarizado con todas las teclas de letras y de puntuación de las tres hileras inferiores del teclado. Pero aún no has tocado las siempre importantes (en los listados de programas) teclas numéricas, ni has aprendido cómo modificar las teclas para obtener letras mayúsculas o minúsculas, así como los símbolos adicionales y de puntuación disponibles.

El programa de esta sección te muestra cómo añadir todo esto a tus crecientes habilidades mecanográficas. Además, hay un ejercicio que te permitirá incrementar tu velocidad, tu seguridad y tu ritmo. Y además es divertido.

ADICION DE LOS NUMEROS

Al igual que con el programa anterior la adición de la línea de números del teclado puede llevarse a cabo con una simple modificación del programa original utilizado en las lecciones anteriores. Para ello, carga el último programa y teclea estas líneas adicionales. Algunas de ellas sustituyen a líneas existentes, mientras que otras van numeradas de modo que quedarán incluidas entre las que ya tienes.

```
  30 LET S$="1A2S3D4F5G6H
     7J8K9L0"
 210 FOR K=6 TO 24
 230 LET R$=S$(K-5)
 320 LET RN=INT (RND*19)+1
 330 PRINT AT 10, RN+5;"*":
     LET R$=S$(RN)
 350 PRINT AT 10,RN+5;" "
 440 LET RN=INT (RND*19)+1
 530 PRINT AT 10,13;"    ":
     PRINT AT 10,13;T$
 540 FOR M=1 TO LEN T$:
     PRINT AT 9,11+M;"
     *      "
 610 FOR N=1 TO 5: RESTORE:
     LET RN=INT (RND*24)+1:
     FOR K=1 TO RN: READ
     X$: NEXT K
1010 PRINT AT 12,6;S$
2000 DATA "MC6809E",
     "VALOR","ULTIMO",
     "Z80A","RELAX","6502",
     "A37XZ","1024",
     "EXTENSO","JUNCO",
     "RETORNAR","67VDG"
2010 DATA "SOLO",
     "EXCITADO","74LS83",
     "VUELTA","VAPOR",
     "1984","30123",
     "ENFADO","4MHZ",
     "WX101","64MB","VIDEO"
```

Al ejecutar el programa, se te pedirá que selecciones uno de los cinco niveles ya familiares. Éstos son similares a los de antes, con la adición de los caracteres nuevos. En los niveles inferiores, se te pedirá que mezcles correctamente números, y algunas veces signos de puntuación, con las teclas existentes. Esto te resultará difícil, garantizando que no puedas concentrarte simplemente en los números.

A continuación, en los niveles superiores, se te pedirá que teclees una mezcla de palabras, grupos de números y grupos formados por una mezcla de letras y números. Verás las palabras y los números seleccionados en las instrucciones DATA próximas al final del programa. Si lo deseas, puedes modificarlas al cabo de cierto tiempo, ya que de lo contrario llegarías a familiarizarte demasiado con lo que el ordenador va a pedirte que hagas y no se trataría ya de un auténtico desafío. Pero recuerda mantener el mismo número de palabras (o de grupos de caracteres) o el ordenador no podrá leer la cantidad correcta de datos.

Cuando domines las teclas de números, habrá llegado el momento de pasar a la siguiente lección. Ésta te proporcionará la práctica necesaria para obtener los caracteres para los cuales se debe pulsar la tecla SHIFT.

LA TECLA SHIFT

Esta vez, las líneas adicionales de programa te darán la posibilidad de utilizar la tecla SHIFT.

Como antes,



dichas líneas sustituirán a las existentes o se incluirán entre ellas:

```
 20 POKE 23658,0: LET ER=0
 30 LET S$="Alas@sD#dF$fG
    %9H&hJ'jK(kL)10"
210 FOR K=2 TO 29
```

```
230 LET R$=S$(K-1)
320 LET RN=INT (RND*28)+1
330 PRINT AT 10,RN+1;"*":
    LET R$=S$(RN)
350 PRINT AT 10,RN+1;" "
440 LET RN=INT (RND*28)+1
610 FOR N=1 TO 4: RESTORE:
    LET RN=INT (RND*24)+1:
    FOR K=1 TO RN: READ
    X$: NEXT K
1010 PRINT AT 12,2;S$
2000 DATA "$235.50",
    "PRINT#", "&H1200",
    "23.5%", "Contabilidad",
    "LONDRES", "25'768",
    "15@\12", "(bajo)",
    "H+9=1D", "**Listar**",
    "Fuego!!"
2010 DATA ";-;;-;", "Extra",
    "Long-Play", "Sera",
    "4*4=16", "Porque?",
    "6:10 pm", "56'492",
    "\15.40", "Disco",
    "ATENCION", "100/4"
```

Ahora, los niveles inferiores del test te presentan todos los caracteres que están disponibles solamente cuando las teclas se pulsan junto con la tecla SHIFT, símbolos de puntuación, matemáticos, etc. Para hacértelo más difícil van mezclados con las letras originales de cada tecla, lo que significa que tienes que hacer ir y venir los dedos cada vez.

En los niveles superiores tienes ahora una lista de palabras y de grupos de palabras al igual que antes, excepto que esta vez encontrarás mayúsculas y símbolos mezclados con minúsculas. Tu ordenador comprobará tu velocidad y tu precisión en dichas palabras y grupos de caracteres. Si ves que lo haces demasiado bien en estos ejemplos concretos, puedes en cualquier momento montarte tú mismo un nuevo conjunto de datos. Sin embargo, debes recordar que hay que conservar igual el número total de datos.

Cuando puedas encontrar todos los caracteres del teclado, sin mirar y sin vacilar en absoluto, estarás en condiciones de pasar a la sección siguiente.



JUEGO DE VELOCIDAD

Es el momento de pensar en incrementar tu precisión y tu velocidad. Uno de los mejores modos de lograrlo consiste en teclear siguiendo los tiempos de un metrónomo, o algo similar, lo que mejorará tu regularidad y tu ritmo. A continuación, a medida que aumente tu destreza, puedes aumentar la rapidez del metrónomo y, por tanto, la de tus pulsaciones.

Pero ¿por qué utilizar un metrónomo si tu ordenador dispone de un reloj incorporado? El siguiente programa es un completo y nuevo ejercicio de mecanografía dispuesto en forma de juego, en el cual tu puntuación dependerá de cuán bueno seas con el teclado. Consta de dos partes. La primera parte visualiza una línea de caracteres seleccionados de modo aleatorio, que tú deberás teclear en el mismo orden en que aparecen. La segunda parte es más difícil, ya que ahora los caracteres aparecen aleatoriamente en la pantalla uno a uno, de modo que no tengas ningún indicio de cuál es el que viene a continuación.

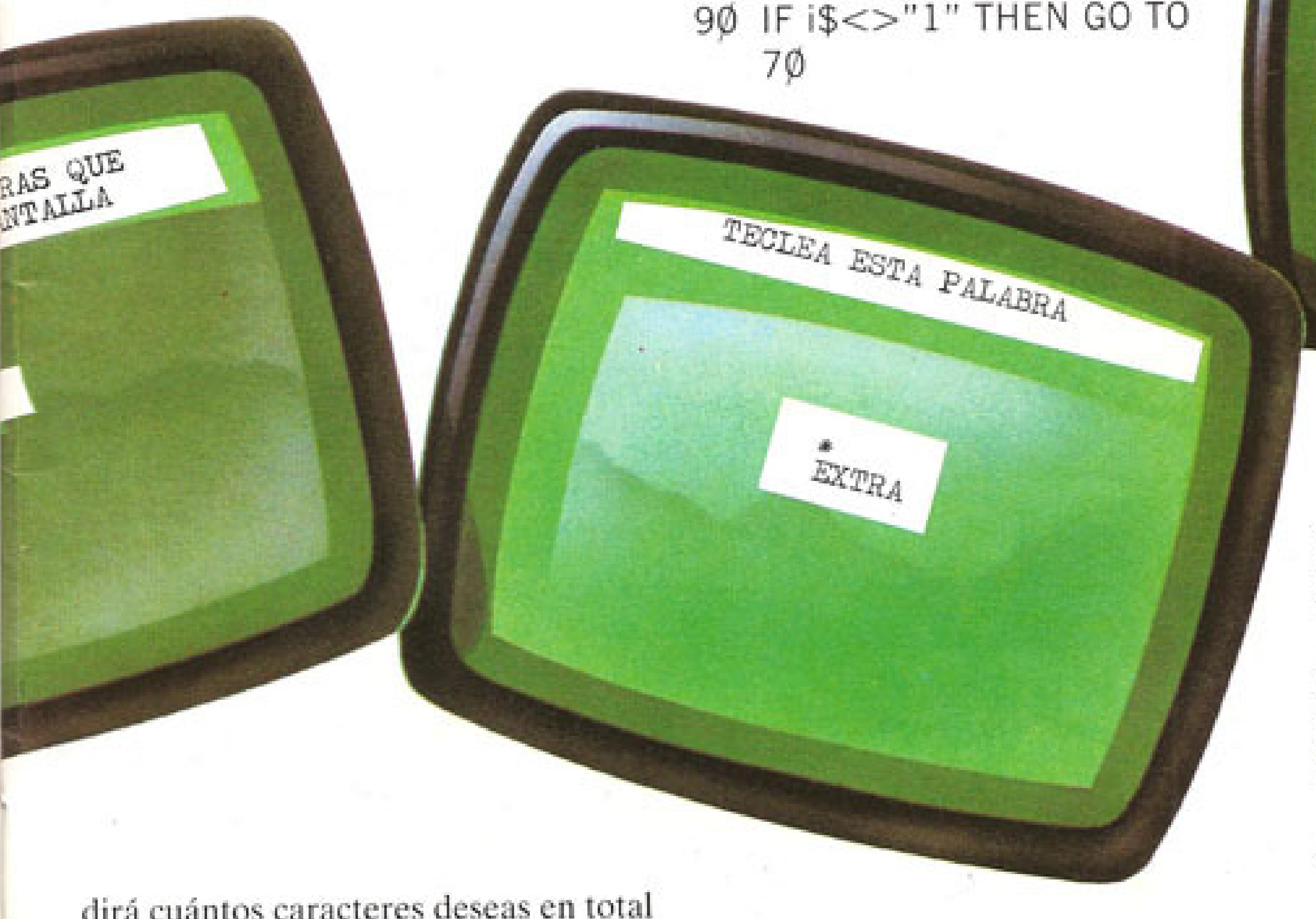
Antes de empezar el test, puedes seleccionar tu propio nivel de dificultad.

Esto se lleva a cabo diciéndole al ordenador a qué velocidad quieres que se visualicen las letras, es decir cuántos caracteres por minuto deseas teclear. Entonces, el ordenador te dará un tiempo limitado en el cual debes teclear cada carácter, dándote en caso contrario un punto de error. En el primer nivel esto se traduce en un indicador móvil que te muestra qué letra deberías estar tecleando, mientras que, en el segundo nivel, es el propio carácter el que se muestra parpadeando solamente durante un intervalo de tiempo determinado.

Antes de que empieces el primer test puedes elegir si deseas el teclado normal (sólo letras) o el teclado expandido (todos los símbolos). Además, antes de que empieces el segundo nivel (el test de caracteres), puedes decidir su duración. Se te pedirá una señal acústica junto al indicador visual correspondiente a cada letra.

Ahora teclea el programa y pon a prueba tu habilidad:

```
10 BORDER 7: PAPER 7: INK
   0: CLS
20 LET a$="ABCDEFGHIJKLM
   NOPQRSTUVWXYZ"
30 LET a$=a$+"abcdefghijkl
   mnopqrstuvwxyz"
40 LET a$=a$+"123456789
   0!@#$%&'()"
50 LET a$=a$+CHR$
   34+"<>;-+=`?/*,."
60 PRINT INVERSE 1;AT 6,7;"
   TEST 1 O TEST 2"
70 IF INKEY$="" THEN GO TO
   70
80 LET i$=INKEY$: IF i$="2"
   THEN GO TO 400
90 IF i$<>"1" THEN GO TO
   70
160 PRINT BRIGHT 1;AT 11,1;
    s$
200 GO SUB 800: LET er=0:
    FOR r=1 TO 30
210 POKE 23672,0: POKE
    23673,0
220 PRINT AT 10,r-1;" *"
230 BEEP .02,20
240 IF PEEK 23672+256*PEEK
    23673>=t THEN LET
    er=er+1: GO TO 300
250 LET i$=INKEY$: IF i$=""
    THEN GO TO 240
260 IF i$=s$(r) THEN PRINT AT
    12,r;"^": GO TO 280
270 LET er=er+1
280 IF PEEK 23672+256*PEEK
    23673<t THEN GO TO 280
300 NEXT r
310 PRINT AT 16,3;"TUVISTES
    ";er;" ERRORES SOBRE 30"
320 FOR f=1 TO 200: NEXT f
330 GO TO 20
400 CLS : INPUT "Numero de
    teclas por minuto? ";cpm
410 INPUT "Numero de
    caracteres? ";r
420 INPUT "Teclas (N)ormal o
    (E)xtendidas? "; LINE m$
430 IF m$="N" OR m$="n"
    THEN LET a$=a$( TO 52):
    GO TO 450
440 IF m$<>"E" AND
    m$<>"e" THEN GO TO 420
450 LET t=3000/cpm
460 GO SUB 800
470 LET er=0
```





dirá cuántos caracteres deseas en total en el test. Cuando el ordenador los haya visualizado todos, se parará y te dará una puntuación basada en tus errores.

No es sólo la velocidad media lo que mide el ordenador en este test, ya que debes mantener un ritmo uniforme. Esto será realmente beneficioso para quien desee incrementar su velocidad general de escritura. Para ayudarte a adquirir la costumbre, el ordenador te

```
100 CLS : INPUT "Cuantos
    caracteres por minuto? ";
    cpm
110 LET t=3000/cpm
120 LET s$=""
130 FOR n=1 TO 30
140 LET s$=s$+a$(INT
    (RND*84)+1)
150 NEXT n
```

```
480Ø FOR n=1 TO r
490Ø POKE 23672,Ø: POKE
     23673,Ø
5ØØ LET r$=a$(INT (RND*LEN
     a$)+1)
51Ø PRINT INVERSE 1;AT 1Ø,
     15;r$; INVERSE Ø;AT 11,
     15;" "
52Ø BEEP .Ø2,2Ø
53Ø IF PEEK 23672+256*PEEK
     23673>=t THEN LET
     er=er+1: GO TO 58Ø
54Ø LET i$=INKEY$: IF i$=""
     THEN GO TO 53Ø
55Ø IF i$=r$ THEN PRINT AT
     11,15;"^": GO TO 57Ø
56Ø LET er=er+1
57Ø IF PEEK 23672+256*PEEK
     23673<t THEN GO TO 57Ø
58Ø NEXT n
59Ø PRINT AT 16,3;"TUVISTE ";
     er;" ERRORES SOBRE ";
     n-1
6ØØ FOR f=1 TO 2ØØ: NEXT f
61Ø GO TO 2Ø
8ØØ LET c$="5..4..3..2..1..Ø"
81Ø FOR n=1 TO 16
82Ø PRINT AT 2,5+n;c$(n)
83Ø PAUSE 1Ø
84Ø NEXT n
85Ø BEEP .2,1Ø
86Ø PRINT AT 2,Ø;TAB 31;" "
87Ø RETURN
```

UN CONSEJO ACERCA DE LA VELOCIDAD

Cuando utilices el programa de velocidad es mejor empezar por el teclado normal a baja velocidad, digamos de 30 a 50 caracteres por minuto.

El riesgo a evitar consiste en teclear los caracteres familiares más rápidamente y luego ir más despacio cuando te encuentres con caracteres sobre los que no estés tan seguro, particularmente si eliges el teclado expandido.

Una vez tengas el hábito de teclear a un ritmo constante puedes seleccionar el teclado expandido y, a continuación, incrementar gradualmente tu velocidad.

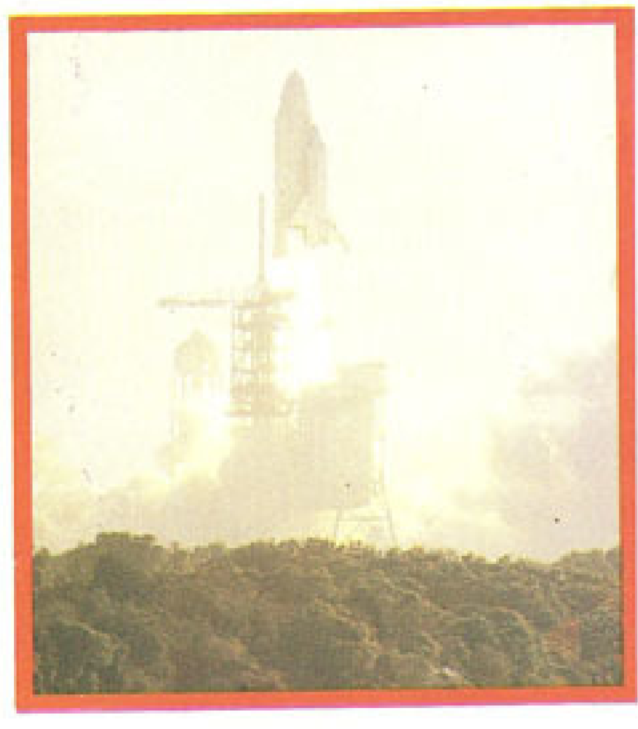
LEJOS DEL MUNDANAL RUIDO

- APUNTAR AL OBJETIVO
- DISTANCIAS
- LA VUELTA AL MUNDO
- MANIOBRAS EN EL ESPACIO
- MOVIMIENTOS PLANETARIOS

Uno de los sueños del hombre ha sido siempre alcanzar las estrellas. Este artículo, complemento del que dedicamos a los objetos voladores, va a enseñarle a tu micro cómo debe conseguir tal objetivo.

En el artículo en que tratamos de trayectorias, «Todo lo que sube baja», publicado en INPUT n.° 12, pudiste ver cómo se puede dividir la velocidad de un proyectil en un componente vertical y otro horizontal. Pudiste ver también cómo se puede modificar el alcance de un proyectil cambiando el ángulo de elevación y la velocidad a la que se disparaba. Este artículo reanuda en este punto el estudio de los cuerpos móviles. Analizará el movimiento de los objetos bajo la ley de gravedad, y te va a permitir conocer su trayectoria desde puntos cercanos a la superficie terrestre hasta muy alejados, hasta quedar en órbita.

Antes de lanzarte al espacio, entra el primer programa, que te demuestra lo útil que nos va a resultar saber algo sobre trayectorias para cuando quieras hacer que tus juegos de proyectiles sean más interesantes y tengan mayor dificultad.



```
10 FOR n=0 TO 31: READ a:
   POKE USR "a"+n,a: NEXT n
20 BORDER 4: PAPER 0: INK
   9: CLS
70 LET a=INT (RND*5): LET
   b=INT (RND*5)+26: LET
   c=INT (RND*8)+2: LET
   h=INT (RND*8)+2: LET
   st=INT (RND*100-c*8)+1:
   LET d=0
90 LET d=d+1
100 CLS : GO SUB 300
110 INPUT "ANGULO?",a2
120 IF a2>89 OR a2<1 THEN
    GO TO 110
130 INPUT "VELOCIDAD?",e
140 IF e=0 THEN GO TO 100
160 LET an=a2*(PI/180): LET
    x=0
170 LET x2=x+(a+1)*8
180 LET y=8+(x*TAN
    an-4*x*x/(e*e*COS an*COS
    an))
185 IF ATTR (21-INT (y/8),INT
    (x2/8))=6 THEN GO TO 245
190 IF y<=0 THEN GO TO 245
200 IF (y>175 OR x2>255) AND
    d<10 THEN GO TO 90
205 IF y>175 OR x2>255 THEN
    GO TO 270
210 PLOT INK 8;x2,y: BEEP .01,
    y/10
220 LET x=x+3
230 GO TO 170
245 IF x2>=b*8+3 AND
    x2<=b*8+10 THEN PRINT
    AT 21,b;CHR$ 145: FOR
    n=20 TO 0 STEP -1: BEEP
    .01,n: NEXT n: GO TO 270
246 IF d<10 THEN GO TO 90
270 IF d=10 THEN PRINT AT 8,
    10;"TIENES QUE
    MEJORAR!": GO TO 280
275 PRINT INVERSE 1;AT 8,10;
    "BUEN TIRO!";AT 10,8;"LO
    CONSEGUISTE EN ";d
280 PAUSE 100: PRINT BRIGHT
    1; PAPER 2; INK 6;AT 13,8;
    "TIENES OTRO JUEGO":
    PAUSE 200
290 GO TO 70
300 PRINT INK 5;AT 21,a;CHR$
    144;AT 21,b;CHR$ 146
310 FOR n=1 TO c: PRINT AT
    21-n+1,12;: FOR m=1 TO
    c: PRINT INK 6;CHR$ 147;:
    NEXT m: NEXT n
```





```
320 RETURN
500 DATA 3,6,60,40,104,60,
    126,255
510 DATA 36,90,165,90,60,
    155,24,60
520 DATA 24,36,66,153,153,
    66,36,127
530 DATA 28,42,85,170,127,
    170,85,255
```

El programa pide que entres la velocidad de despegue y el ángulo de elevación para obtener un disparo que va desde la parte inferior izquierda de la pantalla hasta un objetivo que se halla en la parte inferior derecha. Puedes hacer más difícil el juego colocando el punto de disparo y el objetivo a distancias aleatorias *(random)* para cada serie de intentos, y, todavía más difícil, instalando una barrera de tamaño aleatorio en un punto también aleatorio entre los puntos de partida y de llegada. Cualquiera de las trayectorias que escojas, habrán de tener la suficiente altura como para superar la barrera al efectuarse el disparo.

DISTANCIAS

Este programa es un buen ejemplo de cómo el cerebro humano emite sus juicios. Con sólo mirar las posiciones del fusil, del obstáculo y del objetivo, se puede estimar la velocidad y el ángulo requeridos para obtener una curva que haga que el proyectil se eleve por encima del obstáculo y dé en toda la diana. Con un poco de práctica, vas a sorprender por la cantidad de proyectiles que consigues que den en el objetivo.

Pero medir tales distancias de momento te resultará un tanteo con muchos fallos. Lo que puede ocurrir es que sepas la distancia aproximada del objetivo, y tengas que calcular el ángulo y la velocidad adecuadas. Analizando si el disparo te ha quedado corto o demasiado largo, progresivamente harás cálculos más ajustados.

El siguiente programa muestra cómo hacer esto. Despeja de la memoria el primer programa (CLEAR), pero una vez que lo has guardado (SAVE) si es que deseas volverlo a usar. Entra ahora estas líneas que siguen.

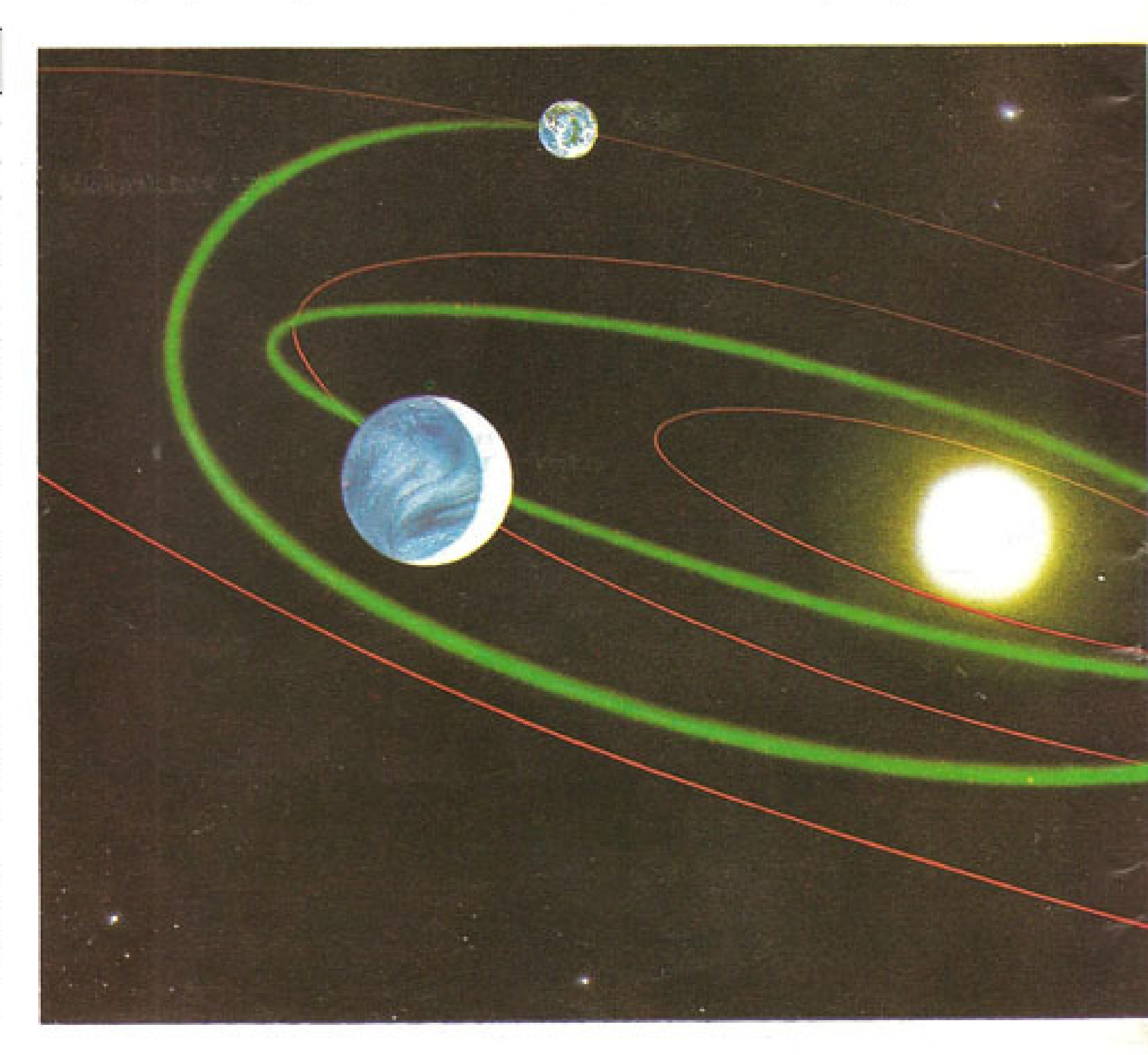
```
10 CLS
20 INPUT "VELOCIDAD
   DISPARO (1-10000 m/s)",
   sp
30 IF sp<1 OR sp>10000
   THEN GO TO 20
40 INPUT AT 4,0;"ANGULO
   DISPARO (1-90 GRA)",a
50 IF a<1 OR a>=90 THEN
   GO TO 40
60 LET a=a*(PI/180)
70 LET r=sp*sp*SIN (2*a)/10
80 PRINT AT 10,3;"EL RANGO
   ES ";INT (r+.5);" metros"
90 PRINT AT 20,1;"PULSA
   UNA TECLA PARA OTRO
   JUEGO (0 FIN)"
100 PAUSE 0: LET a$=INKEY$:
    IF a$="" THEN GO TO 100
110 IF a$<>"0" THEN GO TO
    10
```

Este programa te permite introducir (INPUT) los valores de la velocidad inicial (línea 20) y del ángulo de elevación (línea 40). La línea 70 calcula seguidamente y visualiza el alcance que darán al proyectil estos valores. La variable R es el alcance *(Range)*, SP la velocidad *(SPeed)*, A es el ángulo, y el 10 es el valor aproximado de la gravedad próximos a la superficie terrestre (en realidad es 9,81).

El efecto de la presión del aire no se tiene en cuenta, aunque no debes olvidarla en pruebas que simulen la realidad. La velocidad de disparo más alta que posee un cañón supera los 2.000 m/s. Con ella se han disparado objetos fuera de la Tierra en misión investigadora. Los objetos alcanzan una altura de 180 km.

LA VUELTA AL MUNDO

Si el objeto es disparado con un ángulo que lo eleve por encima de la Tierra, el efecto de la presión es menor, pero si deseas un alcance aún mayor, habrás de tener en cuenta la curvatura de la Tierra. Uno de los primeros cien-

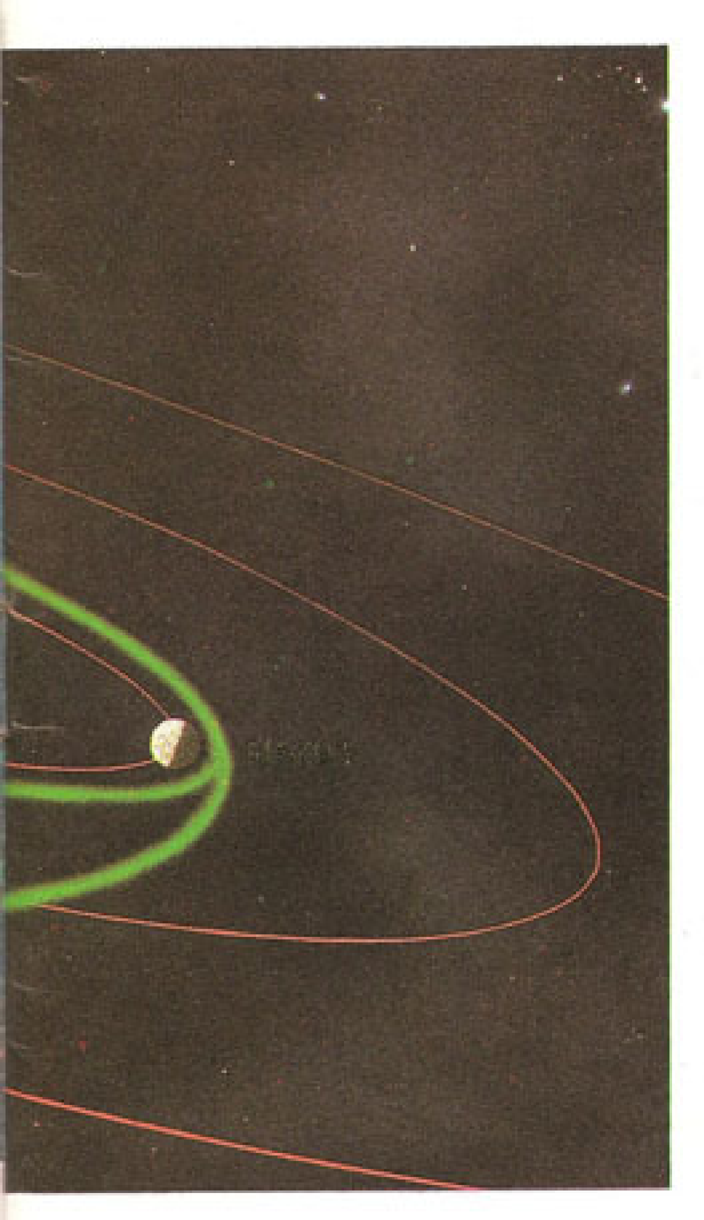


tíficos que se plantearon el problema del máximo alcance de un proyectil que se eleva sobre la superficie de la Tierra fue el británico, sir Isaac Newton.

Newton imaginó un poderoso fusil en la cima de una montaña lo suficientemente alta como para estar fuera de la atmósfera terrestre. Los disparos hechos a una velocidad cada vez mayor llegarían cada vez más lejos si la Tierra fuera plana. Pero dado que nuestro planeta es redondo, la superficie se separa del proyectil, con lo que puede alcanzar una distancia mayor que si la Tierra fuera plana.

Newton argumentaba, además, que se podría entonces disparar un proyectil a tanta velocidad que nunca cayera en tierra, sino que le alcanzara a él por la espalda. Cuando el proyectil fuera a caer sobre la superficie de la Tierra, ésta se curvaría de tal manera que el proyectil quedaría «sin tierra» para siempre: estaría dando vueltas al mundo.

Una vez que cualquier objeto escapa de la gravedad del planeta, su velocidad y distancia respecto del planeta determinan el tipo de órbita que seguirá: circular o elíptica.



Para que se puedan hacer predicciones y medidas sobre un planeta o satélite, debe conocerse su órbita con precisión: hay que saber las características exactas de la elipse. El grado de «achatamiento» de una elipse se denomina *excentricidad* (E). Ésta mide la proporción entre la largura y la anchura de la elipse. Si E es igual a 1, la elipse es igual de ancha que de larga, o sea, es una circunferencia.

Entra y ejecuta (RUN) el siguiente programa para ver el efecto obtenido al variar E con valores que van de cero y pico a mayores que uno:

```
10 CLS
30 INPUT "EXCENTRICIDAD
   (0.1-1.9)",e
40 IF e<.1 OR e>1.9 THEN GO
   TO 30
50 PLOT 127,87+e*40
60 FOR a=0 TO 2*PI+.2 STEP
   .1
70 DRAW 127+(40*SIN
   a)-PEEK 23677,
   87+(e*40*COS a)-PEEK
   23678
80 NEXT a
90 PRINT AT 20,1;"PULSA
   UNA TECLA PARA OTRO
   JUEGO (0 FIN)"
100 PAUSE 0: LET a$=INKEY$:
    IF a$="" THEN GO TO 100
110 IF a$<>"0" THEN GO TO
    10
```

Este programa establece un bucle entre las líneas 30 y 110 para dibujar elipses para las cuales tú introduces (INPUT) el valor de E en la línea 30. Las curvas se dibujan mediante un FOR ... NEXT (líneas 60 a 80) para que dibujen DRAW entre cada punto.

Entra valores de E dentro del intervalo que te muestra la pantalla y comprueba cómo E=1 da una circunferencia, E=0, ... te da elipses aplanadas, y E > 1 te proporciona elipses como huevos de pie. Conclusión: toda órbita no es más que una elipse con su preciso valor de E.

MANIOBRAS EN EL ESPACIO

Una vez en órbita, un satélite o nave espacial no necesita energía alguna para mantenerse en ella, dado que su caída es libre.

Cuanto menor sea el radio de la órbita, más rápidamente se moverá el objeto. Entra el siguiente programa para comprobarlo:

```
10 CLS : LET gc=0
40 LET r=40: LET rt=10+INT
   (RND*60): INF ABS
   (r-rt)<10 THEN GO TO 40
50 CIRCLE 127,87,4
60 LET s=.1: LET a=0: LET
   at=INT (RND*10)+1: LET
   f=0
80 LET a=a+s
100 LET at=at+.1*SQR
    ((40/rt)^3)
110 IF INKEY$="7" AND r<85
    THEN LET f=f+1: LET
    r=r+2: LET s=s*SQR
    (((r-2)/r)^3): GO TO 130
115 IF INKEY$="6" AND r>8
    THEN LET f=f+1: LET
    r=r-2: LET s=s*SQR
    (((r+2)/r)^3): GO TO 130
120 FOR p=1 TO 5: NEXT p
130 LET x=r*SIN a: LET
    y=r*COS a
140 LET xt=rt*SIN at: LET
    yt=rt*COS at
150 PLOT 127+x,87+y
160 BEEP .02,r/2
165 IF gc<>0 THEN PLOT OVER
    1;127+ox,87+oy
170 PLOT OVER 1;127+xt,
    87+yt: LET ox=xt: LET
    oy=yt: LET gc=1
180 IF ABS (x-xt)>4 OR ABS
    (y-yt) THEN GO TO 80
190 PRINT AT 20,12;f;"QUEMA"
200 GO TO 200
```

Ejecútalo (RUN) y observarás la estela de un aparato en órbita y también el satélite objetivo (sin estela). Intenta ajustar la órbita de tu aparato con la del satélite mediante las teclas de flechas de cursor arriba y cursor abajo. La línea 40 establece el radio R de la

órbita del aparato a valor 200 y el radio del objetivo a un valor *random*. La línea 60 establece las variables para las posiciones de partida.

El punto crucial del programa está en la línea 100, que se inspira en otra importante ley física: la raíz cuadrada del tiempo que se tarda en hacer una órbita entera, dividida por el cubo del radio, es constante. Es lo que explica SQR y la potencia a 3 de esta línea.

Para aumentar o disminuir el tamaño de la órbita (hacer maniobras) deberás utilizar las flechas de cursor arriba y abajo. Recuerda que cuanto más pequeñas son las órbitas a más velocidad irás.

MOVIMIENTO PLANETARIO

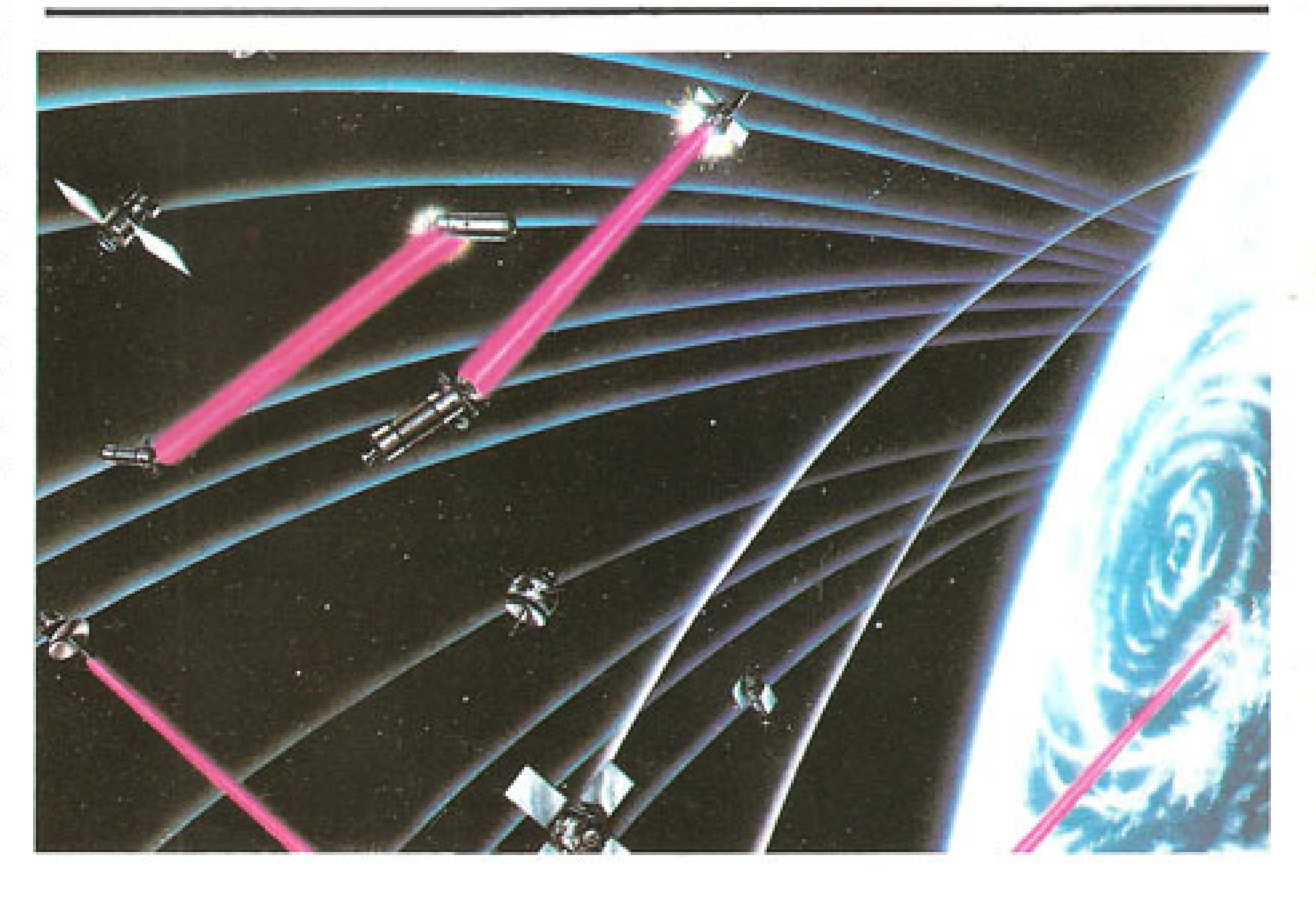
En realidad, las maniobras de salida y entrada a una órbita son mucho más laboriosas que lo que el programa anterior pueda hacer pensar. Es muy fácil trasladarse de una órbita elíptica a otra, pero la inmensidad del espacio hace que sea difícil localizar un objetivo. Y para complicar las cosas interviene de nuevo el efecto de la gravedad. La gravedad del Sol, de la Luna y de los planetas ejerce su influencia sobre la trayectoria del aparato espacial.

En la práctica las herramientas de los astronautas son precisamente unos poderosos ordenadores. Éstos se encargan de controlar la velocidad, el tiempo y la dirección del encendido de los cohetes para mantener la nave en la trayectoria deseada.

Una vez situada en la trayectoria correcta, la nave queda sometida al influjo del campo gravitatorio del sistema solar. Sólo son necesarias unas pocas correcciones en meses y años de viaje: justo cuando los planetas y el Sol se interponen año tras año. El siguiente programa va a permitirte observar los planetas en movimiento:

```
10 BORDER 4: CLS : LET gc=0
20 DIM d(9): DIM p(9): DIM
   i(9): DIM a(9): DIM b(9):
   DIM x(9): DIM y(9)
30 FOR t=1 TO 9: READ d(t),
   p(t): NEXT t
40 INPUT "Cuantos planetas
   (1-9)",s
50 IF s<1 OR s>9 THEN GO
   TO 40
60 LET sc=d(s)/325: LET
   t=p(s)/75
70 PRINT "Hay ";INT t;" dias
   de retraso"'"entre cada
   punto": PRINT #1;
   "PRESION A SPACE PARA
   CONTINUAR"
75 IF INKEY$<>CHR$ 32
   THEN GO TO 75
80 CLS
100 PRINT #1;"PULSA SPACE
   PARA OTRO JUEGO"
110 LET n=1: LET m=1: LET
   g=PI/180
120 FOR q=1 TO s
130 LET r=d(q)/sc: LET a=r: LET
   b=r: LET e=0: LET p=p(q)/t
140 IF q=1 THEN LET e=.2:
   LET b=a*.98
150 IF q=8 THEN LET e=.26:
   LET b=a*.96
160 LET i(q)+360/p
180 LET y=g*i(q): LET x=INT
   (a*(COS y-e)): LET y=INT
   (b*SIN y)
200 IF gc<>0 THEN PLOT
   BRIGHT 0;127+x(q),
   87+y(q)
210 PLOT BRIGHT 1;
   127+a(q)/4,87+b(q)/4: LET
   x(q)=a(q)/4: LET
   y(q)=b(q)/4
240 LET a(q)=x: LET b(q)=y:
   NEXT q
250 LET m=m+t
260 IF INKEY$=CHR$ 32 THEN
   RUN
270 LET qc=1: GO
   TO 120
280 DATA 58,88,108,225,150,
   365,228,687
290 DATA 778,4333,1427,
   10759,2870,30685
300 DATA 4497,60190,5969,
   90741
```

Cuando ejecutes (RUN) este programa, deberás entrar un valor para seleccionar el número de planetas que deseas observar. Cuanto mayor sea el número (entre el 1 y el 9) más complicadas imágenes obtendrás. Para saber qué planetas estás observando, recuerda que el más próximo al Sol es Mercurio, después le sigue Venus, y luego la Tierra, Marte, Júpiter, Saturno, Urano, Neptuno y Plutón. Ejecuta el programa con valores diferentes y observa que las órbitas de dos planetas (Mercurio y Plutón) son claramente elípticas. De hecho todas son elípticas, sólo que algunas tienen una excentricidad muy pequeña y casi se asimilan a circunferencias.



UN DETALLADO EXAMEN

- DIBUJO DE UNA IMAGEN
- DETERMINACION DE LA ESCALA
- AMPLIACION Y REDUCCION
- DETALLES OCULTOS DE LA IMAGEN
- CONFECCION DE DIBUJOS PRECISOS

Este versátil programa puede reducir una imagen a un punto microscópico o ampliarla a dimensiones astronómicas. Utilízalo para crear dibujos altamente precisos o como juego.

El programa que acompaña a este artículo es bastante más divertido que los programas usuales de aplicaciones. De hecho, puede utilizarse como juego o como aplicación.

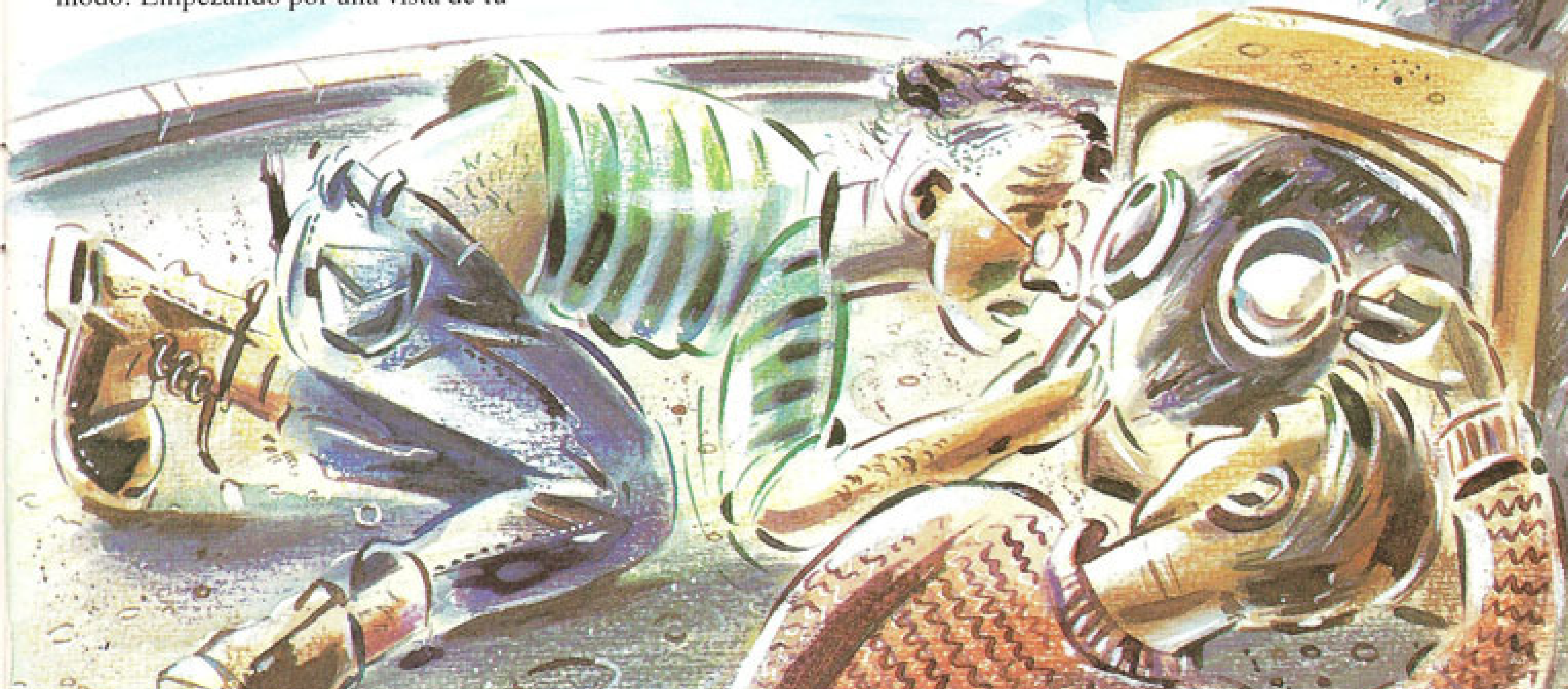
Te permite dibujar una imagen y, a continuación, ampliar cualquier parte de la misma para dibujarla con más detalle hasta una escala microscópica. Puedes cambiar la escala en cualquier momento, utilizando ampliaciones de varios millares o, incluso, centenas de millar. En una imagen, puede quedar oculta una inmensa cantidad de detalles que desaparecen de la vista a medida que te alejas y reaparece de nuevo a medida que te aproximas.

Imagínate con un potente microscopio, asomándote más y más de cerca a alguna parte de la escena. Aparecen los detalles a medida que la ampliación se incrementa y se hacen borrosos hasta desaparecer a medida que decrece. O puedes mirarlo de otro modo. Empezando por una vista de tu casa, puedes imaginar que te mueves más y más lejos perdiéndote en el espacio. Sucesivamente tu casa, tu ciudad, tu país e, incluso, la propia Tierra desaparecen convirtiéndose en un punto en la distancia. Efectos gráficos similares pueden conseguirse utilizando el relativamente corto programa ofrecido aquí.

Un modo de utilizar el programa consiste en convertirlo en un juego para dos personas. Uno dibuja una escena escondiendo algún «tesoro» a una ampliación muy pequeña en algún lugar de la imagen. A continuación, la otra persona debe buscar el tesoro. Puede resultar sorprendentemente difícil. Imagina que empiezas por un cuadrado de 10 centímetros de lado. Si lo amplías 5.000 veces, el cuadrado original se convierte en un área enorme de un cuarto de millón de metros de lado, un espacio suficiente para esconder casi cualquier cosa.

Por supuesto, el programa tiene también aplicaciones serias. Pueden realizarse dibujos muy precisos introduciendo los detalles a una escala mucho mayor que la usual y, a continuación, reduciéndolos al tamaño correcto. Se pueden conseguir dibujos técnicos precisos dibujándolos por partes de ese modo. El programa también es útil en la enseñanza. Por ejemplo, en una lección de geografía, la posición de las ciudades principales puede señalarse en un mapa del país pero a escala mucho más reducida. Con una ampliación normal, sólo serán puntos residuales y sólo aparecerán en detalle si eliges ampliar la posición correcta de la imagen.

Incluso podrías introducir los nombres de las ciudades ya que, aunque no puedas introducir texto con este programa, es algo bastante simple dibujar las letras. Utiliza una escala grande para dibujarlas primero en la posición correcta y, a continuación, redúcelas al tamaño correcto.



COMO UTILIZAR EL PROGRAMA

Cuando ejecutes el programa, se te presentará una pantalla en blanco y un pequeño cursor parpadeante en el centro de la misma. Además, habrá dos números. El de la izquierda te da la escala de la pantalla actual, 1 para empezar, y el de la derecha te da el número de líneas dibujadas. Dicho número está dimensionado a un máximo de 600 en el inicio del programa. El programa está controlado por el teclado y es extremadamente fácil de utilizar.

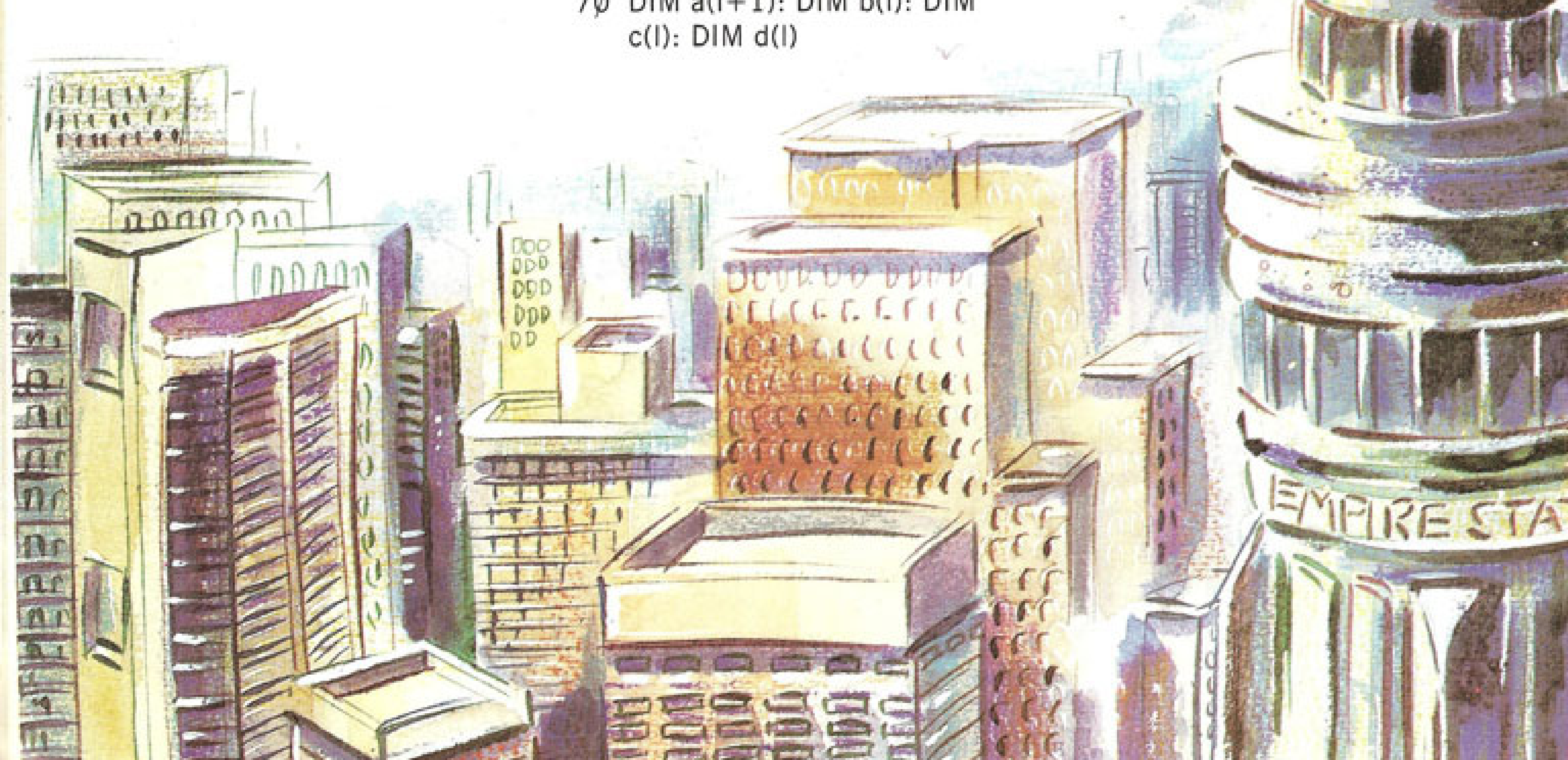
Las teclas del cursor mueven el punto por la pantalla. Puedes acelerar el movimiento del cursor pulsando simultáneamente la tecla SYMBOL SHIFT.

Las otras teclas utilizadas son L, M, C, D, I, O y Z. Esto es lo que hacen. Pulsa la L para dibujar una línea con el cursor o M para mover el cursor sin dibujar una línea. Si pulsas C el cursor vuelve al centro de la pantalla. Si ves que has cometido un error, pulsa D para borrar y se te preguntará cuántas líneas deseas borrar. El programa cuenta hacia atrás a partir de la última línea dibujada y borra el número de líneas correcto. La tecla O obtiene un fichero de salida de modo que puedas salvar tu dibujo, mientras que I carga un fichero correspondiente a un dibujo salvado previamente. Las modificaciones pertinentes para salvar los ficheros en disco se ofrecen después del programa principal.

Ahora vamos por la parte interesante. La tecla Z (zoom) te permite cambiar la escala de la imagen. La escala, o ampliación, se introduce como número. Por ejemplo, 2 dobla la escala y .5 la reduce a la mitad. La imagen se vuelve a dibujar a la nueva escala *centrada en el cursor*, por lo que siempre deberás mover el cursor al punto que deseas ampliar. Si alguna parte de una línea se sale de la pantalla, no se dibuja dicha línea.

Una ampliación a una escala de 0 no hace nada, mientras que una escala de uno vuelve a dibujar la imagen centrada en el cursor, pero a la misma escala. La introducción de un número negativo genera una imagen espejo de la original. Los cambios de escala son acumulativos, por lo que una ampliación de 2 seguida por otra de 2 incrementa la ampliación a 4.

```
10 DEF FN a(x)=(x/256)+128
20 DEF FN b(x)=(x/256)+85
30 BORDER 0: INK 7: PAPER
   0: CLS
50 LET ln=1: LET sc=1: LET
   l=600
60 LET f=0: LET x=0: LET
   y=0: LET zoom=1
70 DIM a(l+1): DIM b(l): DIM
   c(l): DIM d(l)
```






```
80 LET a(1)=x: LET b(1)=y
90 LET c(1)=x: LET d(1)=y
100 LET i$=INKEY$
110 IF i$="l" AND f AND l>ln
    THEN BEEP 0.1,0: PLOT FN
    a(a(ln)),FN b(b(ln)): DRAW
    FN a(c(ln))-PEEK 23677,
    FN b(d(ln))-PEEK 23678:
    LET ln=ln+1: LET a(ln)=x:
    LET b(ln)=y: LET c(ln)=x:
    LET d(ln)=y: LET f=0
120 IF i$="c" THEN LET x=0:
    LET y=0: LET c(ln)=x: LET
    d(ln)=y
130 IF i$="m" THEN LET
    a(ln)=x: LET b(ln)=y
140 IF i$="i" THEN GO SUB
    420
150 IF i$="o" THEN GO SUB
    600
160 IF i$="d" THEN GO SUB
    710
170 IF i$="z" THEN LET lv=0:
    GO SUB 840
180 LET sp=256: IF CODE
    INKEY$<=41 THEN LET
    sp=2048
190 IF INKEY$="8" OR
    INKEY$="(" AND
    x>-32768+sp THEN LET
    f=1: LET x=x+sp: LET
    c(ln)=c(ln)+sp
200 IF INKEY$="5" OR
    INKEY$="%" AND
    x<32768-sp THEN LET
    f=-1: LET x=x-sp: LET
    c(ln)=c(ln)-sp
210 IF INKEY$="7" OR
    INKEY$="'" AND
    y>-22400+sp THEN LET
    f=-1: LET y=y+sp: LET
    d(ln)=d(ln)+sp
220 IF INKEY$="6" OR
    INKEY$="&" AND
    y<22400-sp THEN LET
    f=1: LET y=y-sp: LET
    d(ln)=d(ln)-sp
230 GO SUB 250
240 GO TO 100
250 PRINT AT 1,2; INVERSE 1;"
    ESCALA:";: LET a$=STR$
    sc: IF LEN a$>4 THEN LET
    a$=a$( TO 4)
255 PRINT INVERSE 1;a$;AT 1,
    18;" LINEAS:";ln-1
280 LET bx=FN a(a(ln)): LET
    by=FN b(b(ln)): LET ex=FN
    a(c(ln)): LET ey=FN b(d(ln))
290 PLOT bx,by: PLOT ex,ey
310 PLOT OVER 1;bx,by: PLOT
    OVER 1;ex,ey
320 RETURN
420 CLS
430 INPUT "ENTRA NOMBRE
    FICHERO ";f$
435 IF f$="" THEN GO TO 430
```

```
440 LET ln=1: LET zoom=1:
    LET sc=1
460 LOAD f$ DATA a()
470 LOAD f$ DATA b()
480 LOAD f$ DATA c()
485 LOAD f$ DATA d()
490 LET ln=a(601)
590 RETURN
600 CLS
610 INPUT "ENTRA NOMBRE
    FICHERO ";f$
615 IF f$="" THEN GO TO 610
620 LET lv=1: GO SUB 840
625 LET a(601)=ln
630 SAVE f$ DATA a()
640 SAVE f$ DATA b()
650 SAVE f$ DATA c()
660 SAVE f$ DATA d()
700 RETURN
710 INPUT "ENTRA NUMERO
    DE LINEAS A BORRAR"'k
750 IF k=0 OR ln-k<=0 THEN
    GO TO 830
755 LET ln=ln-k
760 LET x=a(ln): LET y=b(ln)
780 CLS
790 LET c(ln)=x: LET d(ln)=y
800 IF ln=1 THEN GO TO 830
810 IF ABS (c(ln-1))<32767
    AND ABS (d(ln-1))<22399
    THEN LET x=0: LET y=0
820 LET lv=2: GO SUB 840
830 RETURN
840 IF ln=0 THEN RETURN
850 IF lv=1 THEN LET
    zoom=1/sc: CLS : GO TO
    920
860 IF lv=2 THEN LET zoom=1:
    GO TO 920
870 BEEP .1,10: INPUT "ENTRA
    ESCALA ZOOM";
    zoom
890 IF zoom=0 THEN GO TO
    1020
910 CLS
920 FOR u=1 TO
    ln-1
930 LET a(u)=(a(u)-x)*zoom:
    LET b(u)=(b(u)-y)*
    zoom
940 LET c(u)=(c(u)-x)*zoom:
    LET d(u)=(d(u)-y)*
    zoom
950 IF ABS (a(u))<32768 AND
    ABS (b(u))<22400 AND
    ABS (c(u))<32768 AND
    ABS (d(u))<22400 THEN
    PLOT FN a(a(u)),FN b(b(u)):
    DRAW FN a(c(u))-PEEK
    23677,FN b(d(u))-PEEK
    23678
960 NEXT u
970 LET a(u)=(a(u)-x)*zoom:
    LET b(u)=(b(u)-y)*zoom
980 LET c(u)=(c(u)-x)*zoom:
    LET d(u)=(d(u)-y)*zoom
990 LET x=c(ln): LET y=d(ln)
1000 IF ABS (a(ln))>32767 OR
     ABS (b(ln))>22399 THEN
     LET a(ln)=x: LET b(ln)=y
1010 LET sc=sc*zoom
1030 RETURN
```

CONSEJO PARA JUGAR

Cuando utilices tu programa como juego, la parte difícil consiste en asegurarte de que el mensaje oculto o el tesoro no aparezca como una mancha obvia en la imagen.

El truco consiste en dibujar grupos de detalles a distintas ampliaciones, de modo que tu oponente tenga que buscar el área correcta de la imagen así como la ampliación adecuada. Sin embargo, no escondas demasiado bien el tesoro ya que es fácil perderse irremisiblemente en ampliaciones muy grandes.

Si deseas darle al juego una puntuación, simplemente cuenta el número de movimientos y de cambios de escala que lleva a cabo el jugador. El que efectúe el menor número de movimientos es el ganador.

GANADORES DE LOS MEJORES DE INPUT SINCLAIR

En el sorteo correspondiente al número 20 entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

NOMBRE	LOCALIDAD	TITULO ELEGIDO
Pedro M. Pérez González	Málaga	SCOOBY DOO
Íñigo Clemente Lázaro	Calatayud (Zaragoza)	F. MARTÍN
José M. Varela Yebra	León	ARMY MOVES
Antonio Vélez Casellas	Sta. Coloma Gramanet (Barna)	SOCCER
Enrique Cartagena Martínez	Elche (Alicante)	THANATOS
Antonio Rivas Pérez	Vigo (Pontevedra)	COBRA
Ángel Vadillo Rueda	Getafe (Madrid)	F. MARTÍN
Tere Reus Hermoso	Barcelona	TRIVIAL PURSUIT
Jesús Darriba González	Coslada (Madrid)	FIST II
Francesc Arenas Sánchez	Sallent (Barcelona)	ARMY MOVES

RECURSION: BUCLES DENTRO DE BUCLES

Si alguno de tus programas posee subrutinas a las que se llama repetidamente, puedes encontrarte con un caso adecuado para utilizar programación recursiva. Averigua cómo.

Esencialmente, programar un ordenador es un ejercicio de resolución de problemas. A medida que tu habilidad se desarrolle, podrás darte cuenta de que la mayor parte de problemas pueden resolverse si se dividen en problemas menores y más simples. Sin embargo, eventualmente llegarás a una de las pesadillas de los programadores, en la que un intento de resolver un problema te lleva solamente a un nuevo problema, la solución del cual te conduce a otro problema y así sucesivamente.

Puedes considerar que se trata de un punto lo bastante delicado como para hacer un alto e ir en busca de un problema distinto, pero recuerda que tu micro no se ve afectado por esa clase de barrera mental que hace que un cerebro humano aproveche la menor oportunidad para salir de la pesadilla. De hecho, hay una técnica de programación avanzada que sirve para resolver ciertos tipos de problemas que pueden reducirse a problemas dentro de problemas. Dicha técnica se denomina recursión.

UN AIRE DE MISTERIO

Generalmente, los programadores inexpertos ven la recursión como un tema altamente complicado y misterioso. Eso es a causa de que las técnicas de programación utilizadas pueden ser extremadamente difíciles de seguir a partir del listado. Algunas veces, ni siquiera un diagrama de flujo muestra claramente lo que sucede realmente. No obstante, el principio no es difícil de captar.

En términos matemáticos, «recursión» es la repetición de una operación determinada. Sin embargo, esta definición no es estrictamente aplicable a la programación, en donde tiene un significado más preciso. En



esencia, recursión es una llamada primaria a una subrutina o procedimiento con un conjunto de parámetros iniciales. A continuación, la subrutina o procedimiento se llama a sí mismo repetidamente, actualizando los parámetros en cada llamada, hasta que se ha llevado a cabo una determinada tarea.

Para ayudarte a comprender el concepto básico de recursión, piensa en cómo un conductor puede viajar desde el lugar A al lugar B, a través de una gran ciudad con la que no está familiarizado. El conductor dispone de mapas con las calles de la ciudad, pero observa que viajar directamente de A a B es demasiado difícil. Por tanto, decide descomponer el problema en un cierto número de problemas similares (pero mucho más simples), cada uno de los cuales puede irse resolviendo sucesivamente. El modo más sencillo de hacerlo consiste en seleccionar un lugar (C), que se encuentre entre A y B, y decidir cómo viajar desde A hasta C. A continuación conduce su coche hasta C.

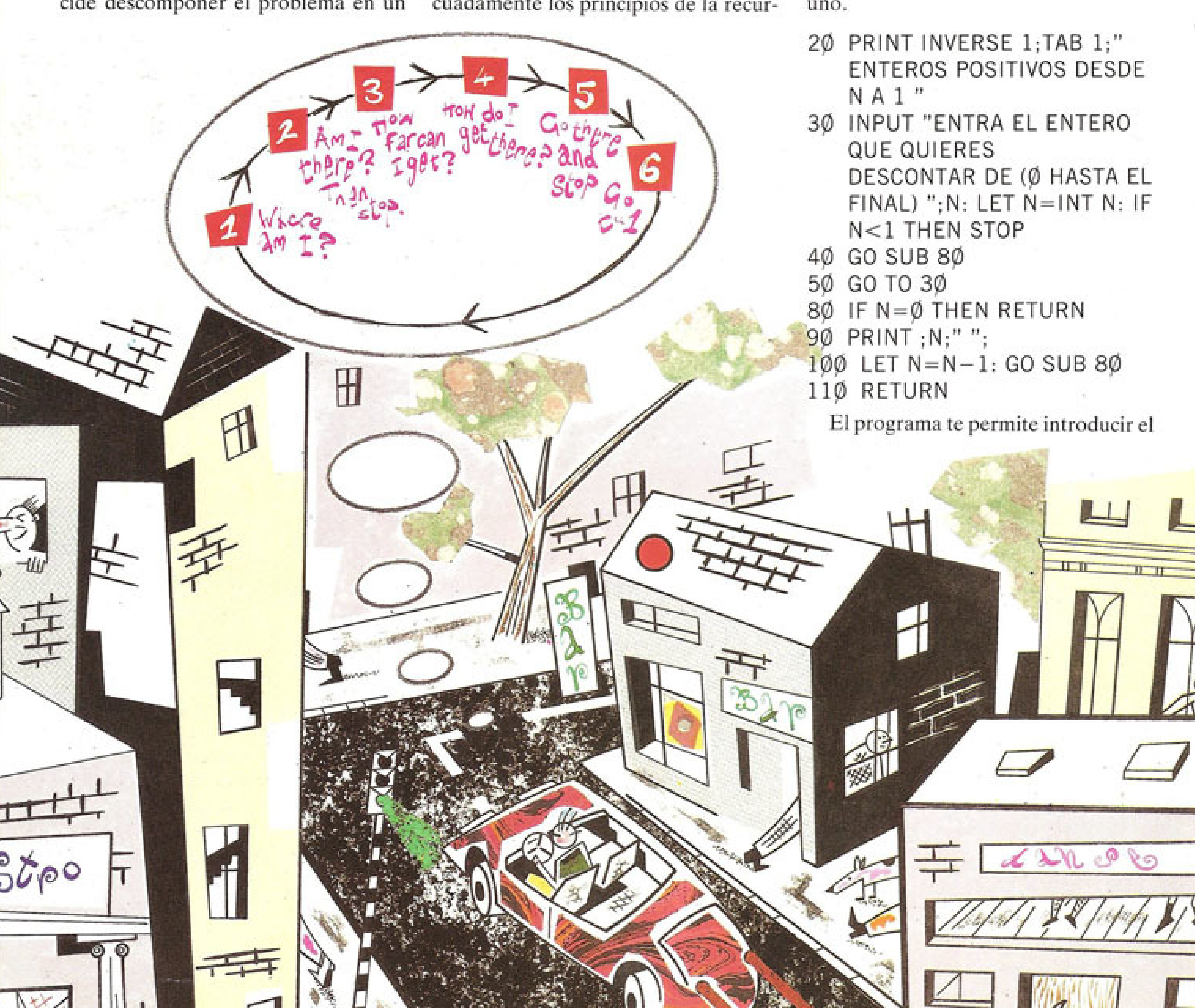
Una vez en C, el conductor mira si puede viajar directamente desde C hasta B. Si es posible, lo hace. En caso contrario, repite el proceso anterior eligiendo un nuevo lugar (D), entre C y B. Este proceso se repite hasta que el conductor llega a B, su punto de destino.

Este ejemplo simple demuestra adecuadamente los principios de la recursión tal como se aplican a la programación de ordenadores.

A medida que se alcanza cada subtarea (o nivel) de la recursión, resulta a menudo necesario almacenar la información relativa a la posición anterior, alcanzada en el nivel precedente, para volver más adelante a ella. A medida que se entra en cada nivel, se obtiene un conjunto de parámetros distinto y se efectúa una comprobación para determinar si se ha realizado la tarea completa. Sin dicha comprobación, el proceso no terminaría nunca. Para ver el método utilizado, introduce y ejecuta el primer programa, que imprime los enteros positivos desde un valor de entrada (N) hasta uno.

```
20 PRINT INVERSE 1;TAB 1;"
   ENTEROS POSITIVOS DESDE
   N A 1 "
30 INPUT "ENTRA EL ENTERO
   QUE QUIERES
   DESCONTAR DE (0 HASTA EL
   FINAL) ";N: LET N=INT N: IF
   N<1 THEN STOP
40 GO SUB 80
50 GO TO 30
80 IF N=0 THEN RETURN
90 PRINT ;N;" ";
100 LET N=N-1: GO SUB 80
110 RETURN
```

El programa te permite introducir el



valor del mayor entero a partir del cual deseas contar. Si introduces un valor menor de uno, se detiene el programa. La línea 40 llama la subrutina recursiva, cuya primera línea comprueba si se ha completado la tarea entera.

Dicha comprobación es crucial para terminar las llamadas recursivas. Se entra en el primer nivel de recursión con el valor de N que has especificado.

Dicho valor se imprime en la línea 90. Se entra en el segundo nivel en la línea 100, que reduce el valor de N en uno y llama la subrutina otra vez con el nuevo valor de N. El programa continúa el bucle entre las líneas 80 y 100, imprimiendo sucesivamente cada entero.

Cuando N se reduce a 0 (en la línea 100), el programa bifurca del modo habitual a la línea 80, en donde esta vez deberá obedecer la instrucción RETURN. Esto provoca un retorno desde la subrutina llamada en la línea 100. La siguiente instrucción se encuentra en la línea 110, que provoca un retorno desde la subrutina llamada en la línea 40. La instrucción siguiente (línea 50) ejecuta de nuevo el programa.

Observa que el programa termina con un valor de N=0, según la línea 100, pero que la línea 90 no imprime nunca dicho valor. Para restaurar N al último valor impreso, puedes teclear LET N=N+1 como línea 105. A continuación, N tomará el último valor impreso.

```
10 DIM N(34): DIM A(34)
20 CLS
30 PRINT TAB 5; INVERSE 1;
   "CALCULO DE
   FACTORIALES"
40 INPUT "ENTRA EL NUMERO
   FACTORIAL QUE QUIERES
   (1-33, O 0 PARA
   TERMINAR)";NU
50 IF NU>33 OR NU<>INT
   (NU) OR NU<0 THEN RUN
60 IF NU=0 THEN STOP
70 LET LE=1: LET N(LE)=NU:
   LET AN=NU
80 GO SUB 150
90 PRINT AN;"! = ";A(1): PRINT
   : GO TO 40
150 IF N(LE)=0 THEN LET
    A(LE)=1: GO TO 180
160 LET LE=LE+1: LET
    N(LE)=N(LE-1)-1: GO
    SUB 150
170 LET LE=LE-1: LET
    A(LE)=A(LE+1)*N(LE)
180 RETURN
```

Ejecuta el programa e introduce un valor en respuesta al mensaje. El programa calcula e imprime el *factorial* del número que hayas introducido, es decir el producto de cada entero desde uno hasta el propio número inclusive. Por ejemplo, 5 factorial (que se escribe 5!) es 1×2×3×4×5=120.

El programa dimensiona variables (línea 10) en número suficiente para completar la tarea. Dichas variables de matriz reservan espacio de memoria para su uso exclusivo, utilizando el nivel de recursión (LE) como subíndice de la variable que se esté utilizando en ese momento, N(LE).

Esencialmente, el programa empieza en la línea 70, en donde el nivel se pone en uno. Además, aquí se asigna el número que hayas introducido, por ejemplo cinco, a N(1) y a AN (la variable que acumula la respuesta). A continuación, la línea 80 llama al primer nivel de recursión. La primera línea de la rutina de recursión



(línea 150) comprueba si se ha llegado al final del problema, cuando N(LE)=0. Pero, por el momento, N(LE) es cinco, por lo que el control pasa a la línea 160. Ésta incrementa el nivel (a dos), pone en 4 el número actual, luego llama de nuevo la rutina recursiva y así sucesivamente. Cuando la línea 160 incrementa el nivel a seis y decrementa el número actual a cero, la línea 150 detecta que N(6) es cero, por lo que el elemento seis de la matriz A se pone a uno y la línea 180 en-

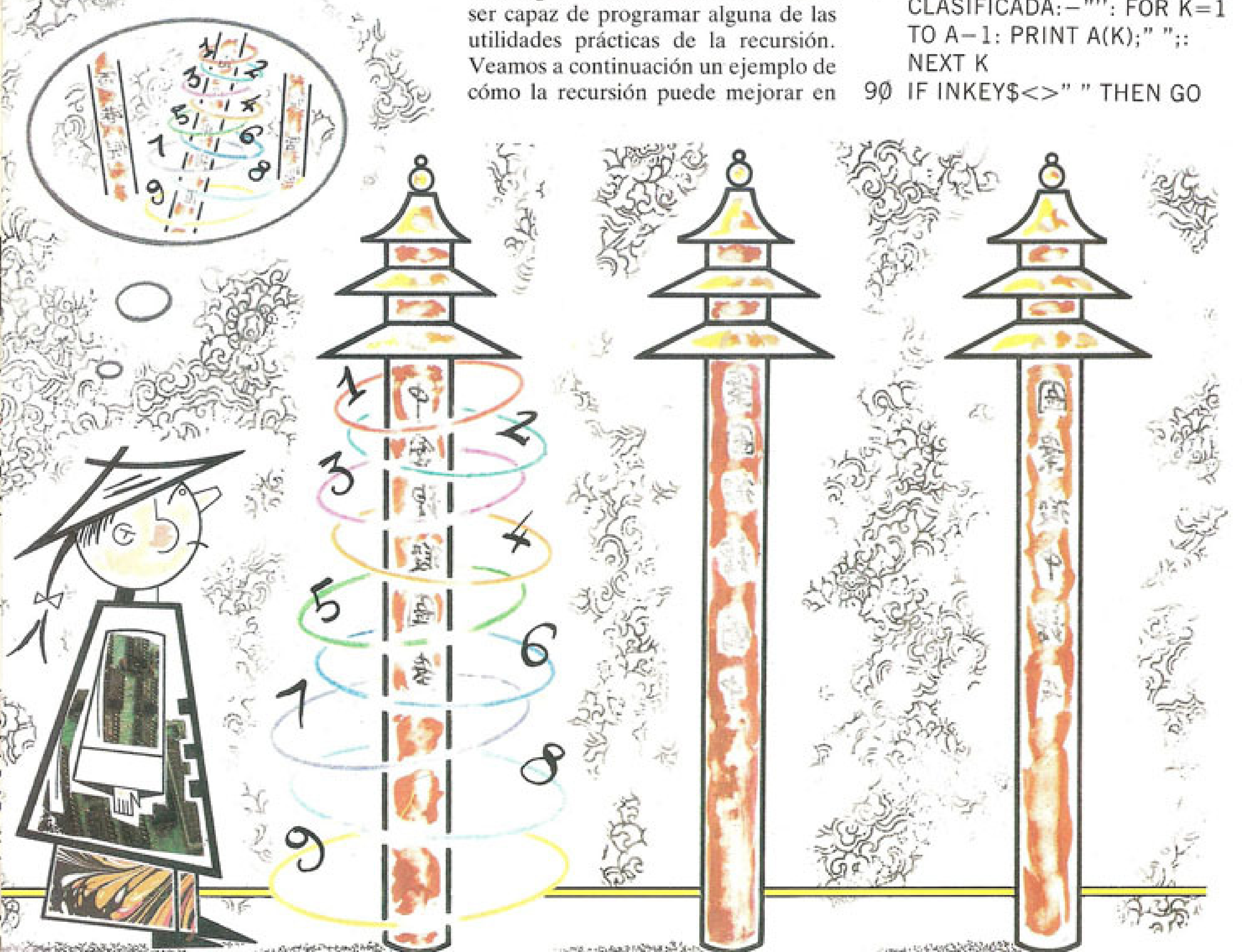
vía un RETURN al final de la línea 160. Ahora el control pasa a la línea 170, en donde el nivel se decrementa a cinco y A(5) toma un valor igual a A(6) veces N(5). Esto hace que A(5) sea igual a 1×1=1. Ahora la línea 180 devuelve de nuevo el control al final de la línea 150 en donde, esta vez, A(4) toma un valor igual a A(5) veces N(4). Por lo que A(4) es igual a uno (calculado antes) por 2. Este bucle se continúa tantas veces como se haya llamado el GOSUB de la línea 160. Cuando se completa el bucle, LE es igual a uno y el último RETURN va a la línea 80. La instrucción siguiente (línea 90) imprime el resultado, 120.

Para mantener tu programa dentro de los límites del micro y evitar catástrofes, debes saber cómo se comporta en el nivel más bajo de recursión. Habitualmente, el único medio de estar seguro de que funciona correctamente consiste en utilizar datos de comprobación de los que estés seguro. Un método simple consiste en considerar una subrutina recursiva como si fuera un grupo de copias similares de la misma subrutina.

Cuando escribas este tipo de subrutinas, empieza siempre por una comprobación de salida efectiva. Dicha comprobación se utiliza para decidir si el problema (tal como viene determinado por los parámetros de entrada) puede resolverse directamente, sin necesidad de subdivisión.

Antes de que empieces a programar, planea lo que quieres conseguir con la subrutina. A continuación, cuando estés escribiendo el programa, no te preocupes demasiado por la secuencia exacta de ejecución, y considera en cambio los dos principios principales, la condición de parada y la subdivisión en problemas más sencillos. Siguiendo este método, deberías ser capaz de programar alguna de las utilidades prácticas de la recursión. Veamos a continuación un ejemplo de cómo la recursión puede mejorar en gran medida la claridad y la eficiencia de un programa de ordenación.

```
10 BORDER 1: INK 7: PAPER 1:
   CLS
20 PRINT TAB 6; INVERSE 1;"
   CLASIFICACION RAPIDA "
30 INPUT "CUANTOS NUMEROS
   DESEAS CLASIFICAR
   (1-1000) ";A
40 IF A<1 OR A>1000 THEN
   GO TO 10
50 DIM A(A): DIM R(2+SQR (A))
60 LET A(A)=100: PRINT
   INVERSE 1;'"TABLA NO
   CLASIFICADA:-"': FOR K=1
   TO A-1: LET A(K)=INT
   (RND*99): PRINT A(K);" ";:
   NEXT K
70 LET L=1: LET LV=1: LET
   R=A-1: GO SUB 1000
80 PRINT INVERSE 1;''"TABLA
   CLASIFICADA:-"': FOR K=1
   TO A-1: PRINT A(K);" ";:
   NEXT K
90 IF INKEY$<>" " THEN GO
```



```
   TO 9Ø
1ØØ RUN
1ØØØ IF R>L THEN LET I=L:
     LET J=R+1: LET V=A(L):
     GO TO 1Ø1Ø
1ØØ5 RETURN
1Ø1Ø LET I=I+1: IF A(I)<V
     THEN GO TO 1Ø1Ø
1Ø2Ø LET J=J-1: IF A(J)>V
     THEN GO TO 1Ø2Ø
1Ø3Ø IF J>=I THEN LET T=A(I):
     LET A(I)=A(J): LET
     A(J)=T: GO TO 1Ø1Ø
1Ø4Ø LET T=A(L): LET
     A(L)=A(J): LET A(J)=T
1Ø5Ø LET R(LV)=R: LET
     LV=LV+1: LET R=J-1:
     GO SUB 1ØØØ
1Ø6Ø LET LV=LV-1: LET
     R=R(LV): LET L=I: GO
     SUB 1ØØØ
1Ø7Ø RETURN
```

El programa te permite introducir la cantidad de números aleatorios que deseas ordenar. La línea 50 dimensiona una matriz A(A), para almacenar dichos números, y otra matriz, R, para almacenar las variables para las llamadas recursivas. La línea 60 genera e imprime los números aleatorios desordenados, mientras que la línea 70 llama la subrutina recursiva que los pone por orden ascendente. La línea 80 imprime la tabla de números.

El método utiliza algunos aspectos de la intercalación de listas a partir de dos subconjuntos y algunos aspectos de la ordenación de sublistas. La lista principal se divide en dos subconjuntos (líneas 1010 y 1020). Observa la comprobación de salida crucial (línea 1000) para determinar cuándo debe terminar la recursión. Cada uno de los dos subconjuntos se ordena en una pasada (línea 1030). Vuelven a dividirse los subconjuntos, pero dos de ellos se juntan formando uno de los nuevos subconjuntos y, a continuación, cada uno de dichos nuevos subconjuntos se ordena en la pasada siguiente. Luego la subrutina se llama a sí misma (línea 1050) repetidamente hasta completar la ordenación.

Aunque este método de ordenación en BASIC no es tan rápido como en código máquina, es extremadamente rápido. Por ejemplo, para ordenar 100 valores tarda alrededor de 40 segundos. Una rutina de ordenación tradicional similar duraría casi una hora.

La utilidad de la recursión llega más allá del cálculo de factoriales y de otras aplicaciones matemáticas. La recursión puede ser extremadamente útil en programas de juegos y para generar figuras gráficas complicadas. También puede aplicarse esta técnica a la inteligencia artificial (AI - Artificial Intelligence), por ejemplo tanto en control de robots como en programas de juegos. Métodos similares se utilizan también en el proceso de lenguajes (compiladores e intérpretes).

En el ajedrez y en los programas de estrategia se utiliza la recursión. Introduce el siguiente programa para ver una ilustración simple de dicha aplicación en el problema clásico de las Torres de Hanoi.

```
1Ø BORDER 6: PAPER 6: INK Ø:
   CLS
2Ø PRINT TAB 8; INVERSE 1;"
   TORRES DE HANOI "
3Ø INPUT "ENTRA NUMERO DE
   ANILLOS (2-9) ";N: IF N<2
   OR N>9 THEN GO TO 3Ø
35 DIM T(3)
36 LET A$="■■": INK 2: FOR
   M=21 TO 21-N STEP -1:
   PRINT AT M,7;A$;AT M,15;
   A$;AT M,23;A$: NEXT M: INK Ø
37 PRINT INK 2;AT 21,7;"■■■";
   AT 21,15;"■■■";AT 21,33;
   "■■■"
38 FOR M=1 TO N: PRINT INK
   7; PAPER Ø;AT 2Ø-T(1),8;
   N+1-M: LET T(1)=T(1)+1:
   NEXT M
39 PRINT #1;AT Ø,5;"UNA
   TECLA PARA COMENZAR"
4Ø PAUSE Ø: PRINT #1;AT Ø,8;
   "          "
45 LET TT=2: LET TF=1: LET
   R=3
5Ø GO SUB 9Ø
7Ø PRINT AT 1Ø,5;"TOTAL DE
   MOVIMIENTOS= ";2^N-1
8Ø STOP
9Ø IF N=Ø THEN RETURN
1ØØ LET N=N-1: LET W=R:
    LET R=TT: LET TT=W: GO
    SUB 9Ø: LET W=R: LET
    R=TT: LET TT=W
11Ø GO SUB 2ØØ
12Ø LET W=R: LET R=TF: LET
    TF=W: GO SUB 9Ø: LET
    W=R: LET R=TF: LET TF=W
13Ø LET N=N+1: RETURN
2ØØ PRINT AT 2Ø-(T(TF)-1),
    TF*8;" "; INVERSE 1;AT
    2Ø-(T(TT)),TT*8;N+1: LET
    T(TF)=T(TF)-1: LET
    T(TT)=T(TT)+1
21Ø BEEP .Ø1,TT*T(TT)*2
22Ø RETURN
```

El problema tradicional consta de tres palos montados sobre una superficie. En el primer palo se ensartan un cierto número de discos de diámetro variable. El objeto consiste en transferir la pila de discos de uno a otro palo. Los discos pueden moverse sólo de uno en uno y ningún disco puede quedar nunca sobre otro menor que él. El tercer palo se utiliza como lugar de paso mientras se mueven los discos. La versión para ordenador da una representación gráfica del problema. Ejecuta el programa e introduce el número de discos que deseas manejar. La transferencia de discos es rápida, por lo que no serás capaz de seguir cada movimiento a menos que modifiques el programa para retardarlo del modo siguiente. Inserta esta línea:

```
215 PAUSE Ø.
```

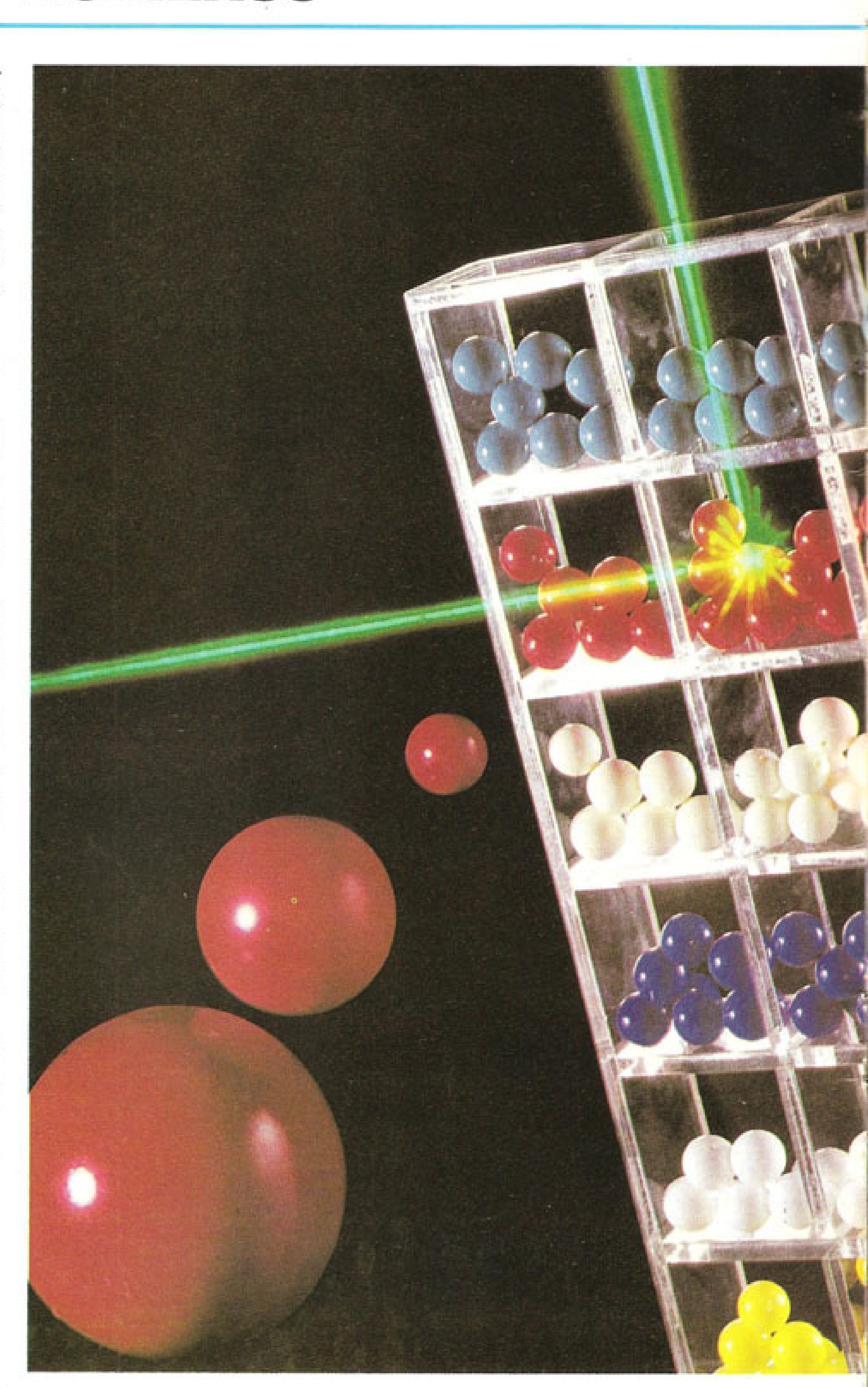
Este cambio hace que el programa se ejecute sólo al pulsar una tecla.

El programa que resuelve el problema está mejor planteado para utilizar la recursión. El formato es similar al de los programas anteriores. La subrutina recursiva imprime sucesivamente cada movimiento de uno a otro palo hasta que se completa el problema. El número total de movimientos efectuados se calcula y se visualiza después de cada jugada.

Por tanto, para problemas altamente complicados que requieran subdivisión, la recursión a menudo es el mejor método de programación.

ROMPECABEZAS DE CUBOS Y NUMEROS

En esta ocasión os presentamos un original juego de ingenio enviado por uno de vosotros. El objetivo a alcanzar es el de colocar en los nueve cuadrados de un lado de un cubo los números del 1 al 9, que no se pueden repetir, de manera que la suma de cada fila y de cada columna coincida con la cantidad que figura en el extremo de cada una de las respectivas direcciones.



Esta cantidad a obtener varía en cada partida que juegues.

Primero se colocan los nueve números, para lo que pulsarás la tecla del número correspondiente y éste aparecerá en el lugar que estaba el interrogante. Después puedes intercambiar los números de dos en dos, invirtiendo su posición respectiva.

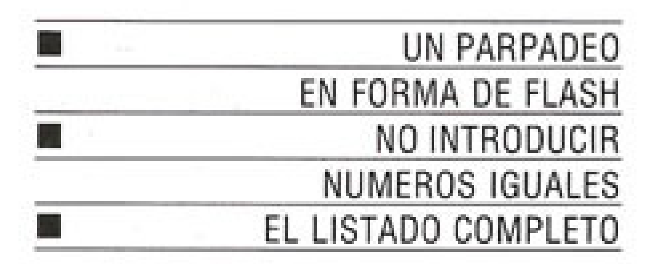
Cuando en alguna de las direcciones sumen la cantidad pedida, la pantalla te lo señalará mediante un parpadeo en forma de *flash*.

Si, por el contrario, no consigues encontrar la solución, el ordenador te la mostrará con tan sólo apretar la tecla «s». En cualquier caso, esto no sucederá hasta que no hayas efectuado un mínimo de diez movimientos.

El ordenador no admitirá jugadas en las que introduzcas dos números iguales y te avisará de tu error. Tampoco contemplará la posibilidad de utilizar teclas que no formen parte de los mandos del juego.

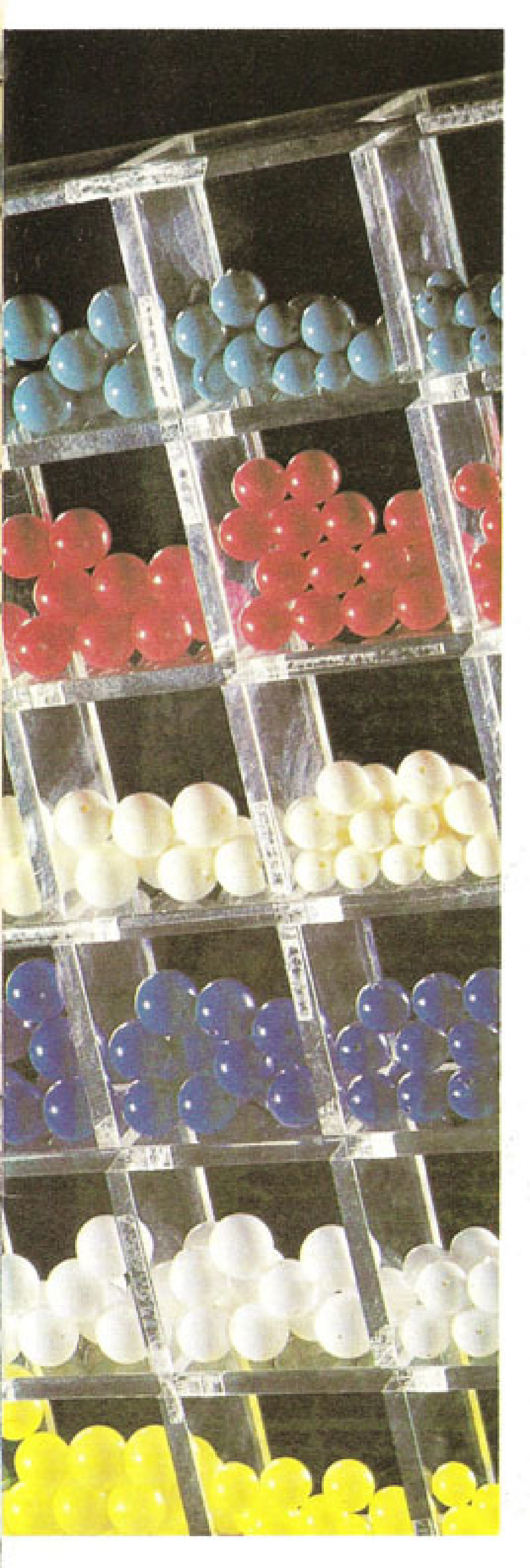
Los números de 1 a 8 que en el listado tengan un guión en la parte superior han de teclearse en modo gráfico y pulsando el número que se indica. En cuanto a los números de 1 a 8 con guión en la parte inferior, han de teclearse en modo gráfico y pulsando las teclas SYMBOL SHIFT y el número que se indica.

```
1 CLEAR 6ØØØØ
3 BORDER 5: PAPER 5: INK 1:
  BRIGHT Ø: FLASH Ø: CLS :
```



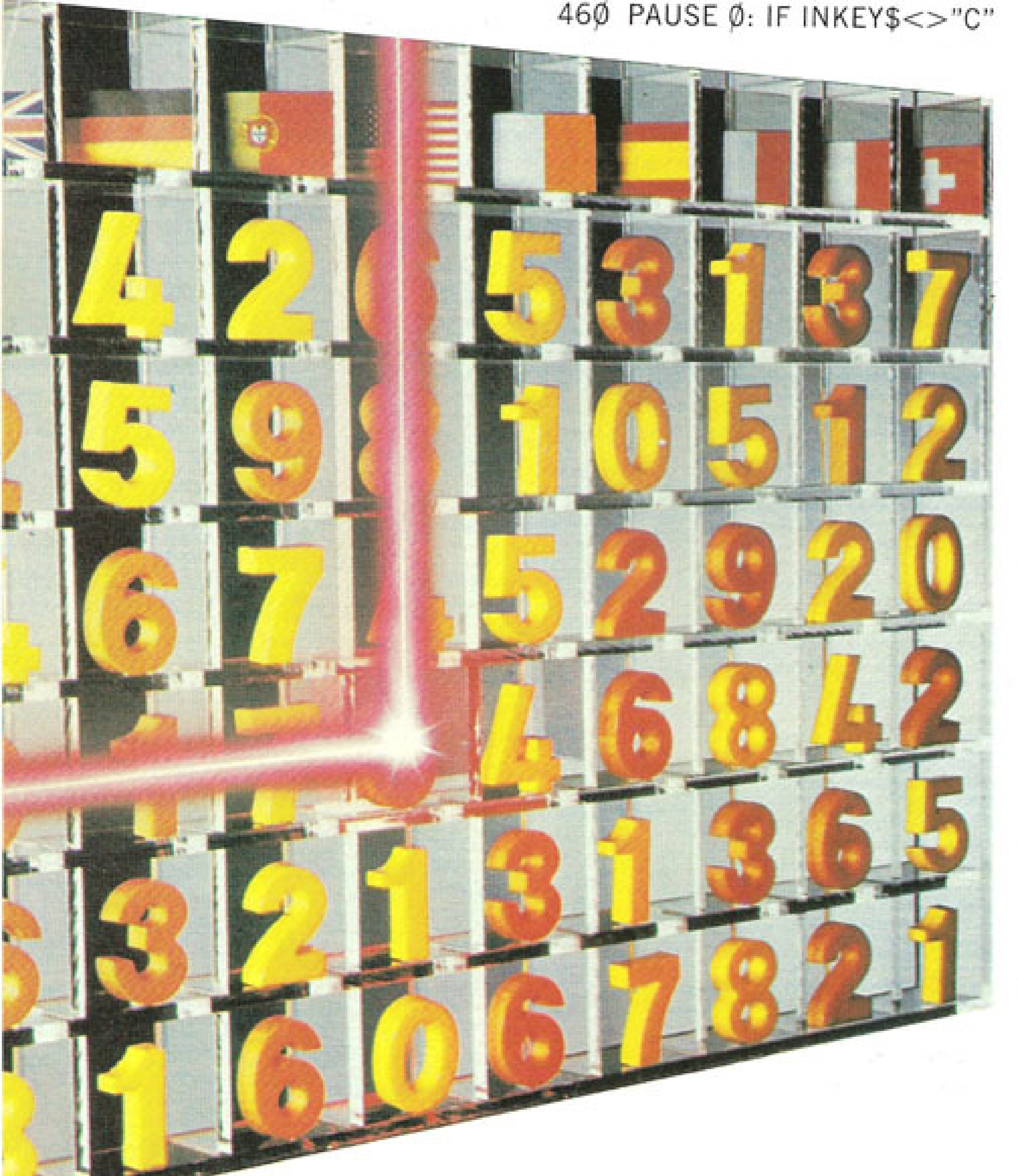
- UN JUEGO DE INGENIO
- NUEVE NUMEROS Y NUEVE CUADRADOS
- EL ORDENADOR CONOCE LA SOLUCION

- UN PARPADEO EN FORMA DE FLASH
- NO INTRODUCIR NUMEROS IGUALES
- EL LISTADO COMPLETO

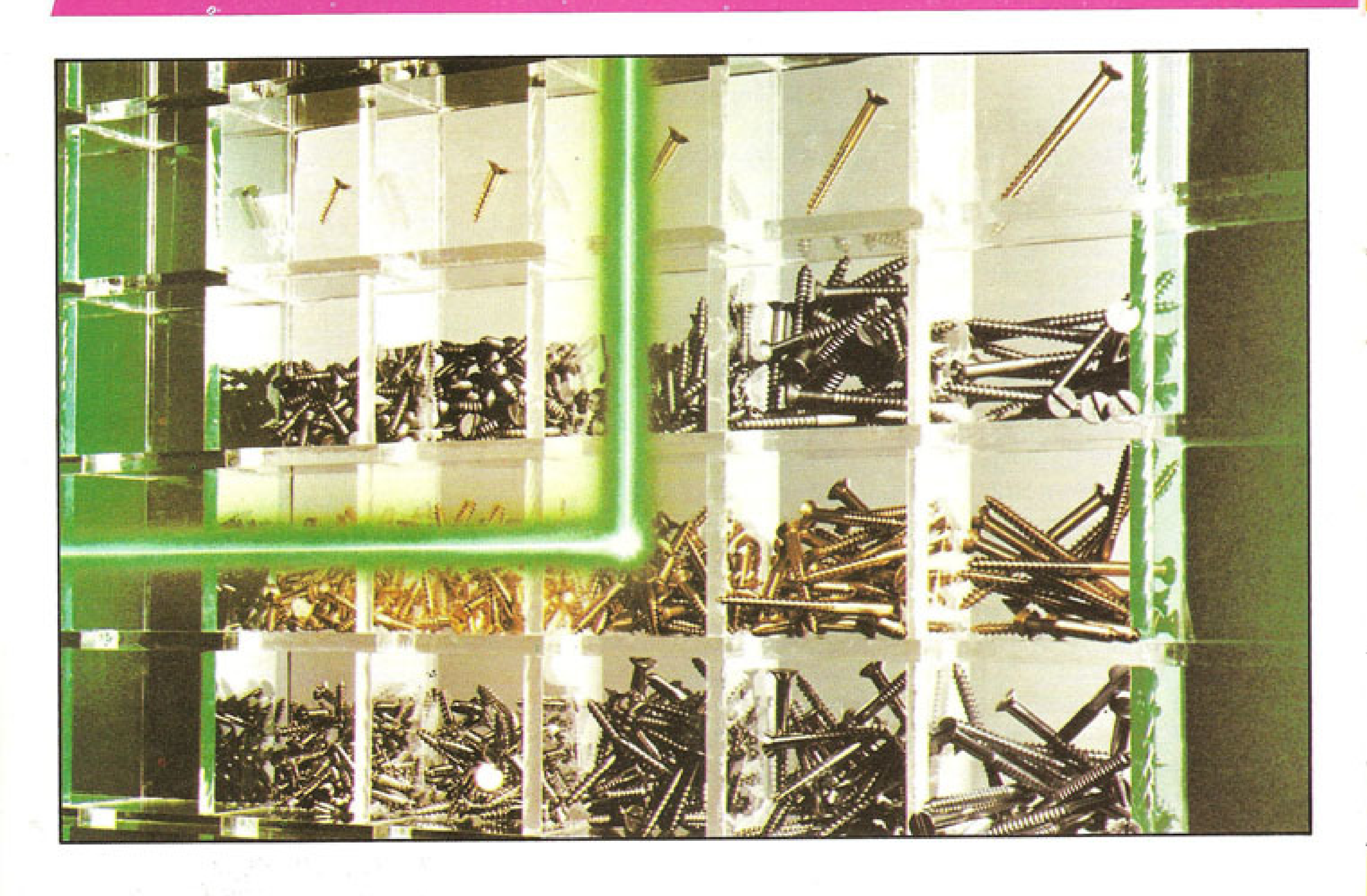


```
   PRINT AT 12,8; PAPER 7;
   FLASH 1;"ESPERA, POR
   FAVOR"
 5 POKE 23658,8: GO SUB
   9550: GO SUB 9800: GO
   SUB 5000
25 REM DIBUJO DE LOS
   NUMEROS *
30 DIM N$(40,2): LET
   N$(1)="7": LET N$(2)="5":
   LET N$(3)="5": LET
   N$(4)="2"
35 LET N$(5)="33": LET
   N$(6)=" 32": LET
   N$(7)="5": LET N$(8)="33"
40 LET N$(9)="33": LET
   N$(10)="42": LET
   N$(11)="5": LET
   N$(12)="33"
45 LET N$(13)="7": LET
   N$(14)="54": LET
   N$(15)="37": LET
   N$(16)="1"
50 LET N$(17)="33": LET
   N$(18)="13": LET
   N$(19)="5": LET
   N$(20)="33"
55 LET N$(21)="33": LET
   N$(22)="13": LET
   N$(23)="55": LET
   N$(24)="33"
60 LET N$(25)="33": LET
   N$(26)="6": LET
   N$(27)="42": LET
   N$(28)="1"
65 LET N$(29)="33": LET
   N$(30)="12": LET
   N$(31)="55": LET
   N$(32)="33"
70 LET N$(33)="33": LET
   N$(34)="55": LET
   N$(35)="37": LET
   N$(36)="33"
75 LET N$(37)="33": LET
   N$(38)="22": LET
   N$(39)="2": LET
   N$(40)="2"
100 REM PRESENTACION
110 BORDER 3: PAPER 3: INK
    0: BRIGHT 0: CLS
120 PRINT AT 0,0; PAPER 0;
    INK 7; FLASH 1;"6666666
    6666666666666666666
    66666"
130 FOR N=1 TO 21: PRINT AT
    N,0; PAPER 0; FLASH 1;
    INK 7;"6";AT N,31;"6":
    NEXT N
150 PRINT AT 6,8; PAPER 7;  "
    43 77 77 743";AT 7,8;   "
    63 554 6 555";AT 8,8;   "
    33 26 325 5435";AT 9,8; "
    ";AT 10,8;              "
    536433";AT 11,8;        "
    5553;AT 12,8;           "
    132 33"
160 PRINT AT 13,8; PAPER 7; "
    37773343437";AT 14,8;   "
    5551325563";AT 15,8;    "
    137632132632332";AT
    16,8;                   "
                "
170 INK 6
180 FOR N=0 TO 14 STEP 2:
    PLOT 64+N,128+N: DRAW
    120,0: DRAW 0,-88: NEXT
    N
200 FOR N=0 TO 30 STEP 2:
    BEEP .1,N+25: NEXT N
240 PRINT AT 19,1; PAPER 7;
    INK 1; FLASH 1;" MARIA
    LUISA CUERVO HERRERO "
250 PRINT AT 21,1; PAPER 1;
    INK 6; FLASH 1;"
    INSTRUCCIONES O JUGAR ?
    (I/J)"
260 PAUSE 0: LET O$=INKEY$:
    IF O$="I" OR O$="i" THEN
    GO TO 300
270 IF O$="J" OR O$="j"
```

```
    THEN GO TO 61Ø
28Ø GO TO 26Ø
3ØØ REM INSTRUCCIONES
31Ø PAPER 5: INK 1: BORDER
    5: BRIGHT Ø: CLS
33Ø PRINT AT 1,1Ø; PAPER Ø;
    INK 4; FLASH 1;
    "INSTRUCCIONES"
35Ø PRINT AT 3,Ø;"COLOCA LOS
    9 NUMEROS (DISTINTOS Y
    DEL 1 AL 9) DE FORMA
    QUE CADA FILA
    (HORIZONTAL VERTICAL Y
    DIA-GONAL) SUME LA
    CANTIDAD QUE SE INDICA
    AL LADO DE DICHA FILA ."
36Ø PRINT "PRIMERO SITUA
    LOS 9 NUMEROS,
    PUL-SANDO EL NUMERO
    QUE QUIERAS PO- NER EN
    EL LUGAR DEL SIGNO DE
    IN-TERROGACION ."
37Ø PRINT "LUEGO
    INTERCAMBIA LOS
    NUMEROS DE2 EN 2,
    INDICANDO LOS QUE
    QUIERAS CAMBIAR .UNA
    LUZ INTERMITENTE INDICA
    LA DIRECCION CUYOS
    NUME- ROS SUMAN EL
    VALOR ADJUNTO."
38Ø PRINT "SI TE
    DESESPERAS,
    PULSA""S""(CUAN-DO
    APAREZCA EL MENSAJE
    OPORTUNO) Y... VERAS LA
    SOLUCION."
45Ø PRINT AT 21,13; PAPER Ø;
    INK 4; FLASH 1;"PULSA C"
46Ø PAUSE Ø: IF INKEY$<>"C"
    AND INKEY$<>"c" THEN
    GO TO 46Ø
47Ø GO TO 1ØØ
61Ø CLS : RANDOMIZE
62Ø REM PREPARAR UNA SERIE
    DISTINTA
63Ø DIM R(9): DIM S(9): DIM
    T(8)
64Ø FOR N=1 TO 9
65Ø LET R=1+INT (9*RND): IF
    R(R) THEN GO TO 65Ø
66Ø LET R(R)=R: LET S(N)=R:
    NEXT N
67Ø FOR N=1 TO 3: LET
    T(N)=S(3*N-2)+S(3*N-1)
    +S(3*N): NEXT N
675 FOR N=1 TO 3: LET
    T(N+3)=S(N)+S(N+3)+S
    (N+6): NEXT N
68Ø LET T(7)=S(1)+S(5)+S(9):
    LET T(8)=S(3)+S(5)+S(7)
75Ø REM DIBUJO DEL TABLERO
76Ø PAPER Ø: INK 6: BORDER
    Ø: BRIGHT 1: CLS
78Ø FOR X=Ø TO 1Ø STEP 5
79Ø FOR M=Ø TO 1Ø STEP 5:
    FOR N=Ø TO 3
8ØØ PRINT AT 2+N+X,3+M;
    "8888"
81Ø NEXT N: BEEP .2,
    3Ø+RND*2Ø: NEXT M:
    NEXT X
85Ø FOR M=Ø TO 4Ø STEP 4Ø:
    FOR N=Ø TO 7 STEP 2
86Ø PLOT 24+N+M,16Ø+N:
    DRAW 32,Ø: DRAW Ø,-32:
    DRAW 7-N,Ø
88Ø PLOT 32+N+M,168+N:
    DRAW 32,Ø: DRAW Ø,-8:
    NEXT N: NEXT M
9ØØ FOR N=Ø TO 15 STEP 2:
    PLOT 1Ø4+N,16Ø+N:
    DRAW 32,Ø: DRAW Ø,-32:
    NEXT N
91Ø FOR N=Ø TO 7 STEP 2:
    PLOT 24+N,12Ø+N: DRAW
    32,Ø: DRAW Ø,-32: DRAW
    7-N,Ø: NEXT N
92Ø FOR N=Ø TO 7 STEP 2:
    PLOT 64+N,127: DRAW Ø,
    N-7: DRAW 32,Ø: DRAW Ø,
    -32: DRAW 7-N,Ø: NEXT N
```



```
930 FOR N=0 TO 7 STEP 2:
    PLOT 104+N,127: DRAW 0,
    N-7: DRAW 32,0: DRAW 0,
    -32: DRAW 7-N,0: PLOT
    136+N,128+N: DRAW 8,0:
    DRAW 0,-32: NEXT N
950 FOR N=0 TO 7 STEP 2:
    PLOT 24+N,80+N: DRAW
    32,0: DRAW 0,-32: NEXT
    N
960 FOR N=0 TO 7 STEP 2:
    PLOT 64+N,87: DRAW 0,
    N-7: DRAW 32,0: DRAW 0,
    -32: NEXT N
970 FOR N=0 TO 7 STEP 2:
    PLOT 104+N,87: DRAW 0,
    N-7: DRAW 32,0: DRAW 0,
    -32: PLOT 136+N,88+N:
    DRAW 8,0: DRAW 0,-32:
    NEXT N
980 FOR N=20 TO 60 STEP 3:
    BEEP .1,N: NEXT N
982 FOR N=1 TO 3: PRINT AT
    5*N-2,0; INK 7;T(N): IF
    T(N)<10 THEN PRINT AT
    5*N-2,0; INK 7;" ";T(N)
984 NEXT N: FOR N=1 TO 3:
    PRINT AT 17,5*N-1; INK
    7;T(N+3): NEXT N
987 PRINT AT 17,17; INK 7;
    T(7): IF T(7)<10 THEN
    PRINT AT 17,17; INK 7;" ";
    T(7)
990 PRINT AT 17,0; INK 7;T(8):
    IF T(8)<10 THEN PRINT AT
    17,0; INK 7;" ";T(8)
1000 PRINT AT 4,21; PAPER 5;
     INK 1;"JUGADA:";AT 11,
     22; PAPER 4; INK 0;
     "NUMEROS";AT 12,20;
     "DISPONIBLES:"
1020 FOR X=1 TO 9: PRINT AT
     14+INT ((X-1)/5)*2,
     21+INT ((X-1)/5)+(X-INT
     ((X-1)/5)*5-1)*2; PAPER
     4; INK 0;X: BEEP .02,
     35+20*RND: NEXT X
1025 PRINT AT 16,2; INK 5;"
     65556";AT 3,2;"3";AT
     8,2; 3;AT 13,2;"3"
1030 BEEP 1,40
1100 REM COLOCAR LOS 9
     NUMEROS
1110 DIM N(9)
1150 FOR N=1 TO 9: BEEP .2,
     30+30*RND
1160 PRINT AT 20,7; PAPER 3;
     INK 6;"NUMERO ?-:";AT
     4,28; PAPER 5; INK 1;N
1180 FOR X=0 TO 3: PRINT AT
     2+X+INT ((N-1)/3)*5,
     (N-INT
     ((N-1)/3)*3)*5-2; FLASH
     1; PAPER 5; INK 1;" ";
     N$(37+X);" ": NEXT X
1200 PAUSE 0: LET
     A$=INKEY$: IF CODE
     A$>57 OR CODE A$<49
     THEN GO TO 1200
1210 LET A=CODE A$-48
1215 BEEP .1,28: PRINT AT 20,
     17; PAPER 3; INK 6;A:
     FOR M=1 TO N-1
1220 IF N(M)=A THEN PRINT
     AT 21,4; PAPER 7; INK 2;
     FLASH 1;"YA ESTA
     COLOCADO": BEEP .7,35:
     PAUSE 60: BEEP .5,48:
     PRINT AT 21,4; INK 0;"
     8888888888888888";
     AT 20,17;"8": BEEP .2,
     23: GO TO 1200
1230 NEXT M: LET N(N)=A
1250 FOR X=0 TO 3: PRINT AT
     2+X+INT ((N-1)/3)*5,
     (N-INT
     ((N-1)/3)*3)*5-2; PAPER
     6; INK 2;" ";
     N$(A*4-3+X);" ": NEXT X
1270 BEEP .3,30+30*RND:
     PRINT AT 14+INT
     ((A-1)/5)*2,21+INT
     ((A-1)/5)+(A-INT
     ((A-1)/5)*5-1)*2; INK 0;
     " "
1290 PRINT AT 20,7; INK 0;"
     8888888888"
1300 NEXT N
1320 PRINT AT 11,22; INK 0;"
     8888888";AT 12,20;
     "888888888888
1330 LET KK=9
1500 REM COMPROBACION
1520 DIM C(8)
1530 FOR N=1 TO 3: IF
     N(3*N-2)+N(3*N-1)+
     N(3*N)=T(N) THEN PRINT
     AT 5*N-2,2; INK 4;
     FLASH 1;" ": LET C(N)=1
1540 NEXT N: FOR N=1 TO 3:
     IF N(N)+N(3+N)+
     N(N+6)=T(N+3) THEN
     PRINT AT 16,5*N-1; INK
     4; FLASH 1;" ": LET
     C(N+3)=1
1550 NEXT N: IF
     N(1)+N(5)+N(9)=T(7)
     THEN PRINT AT 16,17;
     INK 4; FLASH 1;" ": LET
     C(7)=1
1560 IF N(3)+N(5)+N(7)=T(8)
     THEN PRINT AT 16,2; INK
     4; FLASH 1;" ": LET
     C(8)=1
1650 FOR N=1 TO 8: IF C(N)=0
     THEN GO TO 1700
1660 NEXT N: GO TO 3500
1700 REM INTERCAMBIAR 2
     NUMEROS
1710 LET KK=KK+1: PRINT AT
     4,28; PAPER 5; INK 1;KK
1720 BEEP .2,25+25*RND:
     PRINT AT 20,0; PAPER 7;
     INK 1;"CUALES
     INTERCAMBIARAS ?-:"
1730 PRINT AT 8,21; PAPER 3;
     INK 7; FLASH 1;"PULSA
     S,Y";AT 9,21;"VERAS LA
     ";AT 10,21,"SOLUCION."
1750 PAUSE 0: LET
     H$=INKEY$: IF H$="S"
     OR H$="s" THEN GO TO
     4000
1755 IF CODE H$<49 OR CODE
     H$>57 THEN GO TO 1750
1760 LET H=CODE H$-48:
     BEEP .1,30: PRINT AT 20,
     25; PAPER 7; INK 1;H$;
     AT 20,27;"Y"
1790 PAUSE 0: LET
     G$=INKEY$: IF G$="S"
     OR G$="s" THEN GO TO
     4000
1795 IF CODE G$<49 OR CODE
     G$>57 THEN GO TO 1790
1800 LET G=CODE G$-48:
```



```
      BEEP .1,45: PRINT AT 2Ø,
      29; PAPER 7; INK 1;G$
182Ø IF H=G THEN PRINT AT
      21,6; PAPER 5; INK 2;"TE
      HAS CONFUNDIDO": BEEP
      .5,35: PAUSE 6Ø: BEEP
      .4,5Ø: BEEP .3,45: PRINT
      AT 2Ø,25; INK Ø;"88888";
      AT 21,6;"888888888888
      888888": GO TO 175Ø
183Ø PRINT AT 8,21; INK Ø;"88
      8888888";AT 9,21;"8888
      88888";AT 1Ø,21;"8888
      88888"
184Ø PRINT AT 16,2; INK 5;
      "655586  ";AT 3,2;"3";AT
      8,2;"3";AT 13,2;"3"
188Ø FOR N=1 TO 9: IF N(N)=H
      THEN LET J=N
189Ø IF N(N)=G THEN LET F=N
19ØØ NEXT N
191Ø FOR X=Ø TO 3: PRINT AT
      2+X+INT ((F-1)/3)*5,
      (F-INT ((F-1)/3)*3)*5-2;
      PAPER 6; INK 2;" ";
      N$(H*4-3+X);" ": NEXT X
192Ø BEEP .2,25+3Ø*RND:
      FOR X=Ø TO 3: PRINT AT
      2+X+INT ((J-1)/3)*5,
      (J-INT ((J-1)/3)*3)*5-2;
      PAPER 6; INK 2;" ";
      N$(G*4-3+X);" ": NEXT X
193Ø BEEP .2,25+3Ø*RND:
      PRINT AT 2Ø,Ø; INK Ø;"88
      8888888888888888888
      888888888"
194Ø LET N(J)=G: LET N(F)=H
195Ø GO TO 15ØØ
35ØØ REM FINALIZACION
352Ø PRINT AT 16,21; PAPER
      2; INK 7; FLASH 1;
      "CORRECTO!!": FOR N=Ø
      TO 28 STEP 2: BEEP .1,
      2Ø+N: NEXT N
3525 PRINT AT 2Ø,Ø;"
                                  "
353Ø FOR N=1 TO 7:
      RANDOMIZE USR 63ØØ1:
      BEEP .Ø3,2Ø+3*N: NEXT
      N: PRINT AT 21,2; PAPER
      5; INK 1; FLASH 1;"PARA
      JUGAR OTRA VEZ, PULSA
      Y"
355Ø PAUSE Ø: IF
      INKEY$<>"Y" AND
      INKEY$<>"y" THEN GO
      TO 355Ø
356Ø GO TO 1ØØ
4ØØØ REM VISUALIZAR LA
      SOLUCION
4Ø1Ø PRINT AT 2Ø,Ø;"
                                  "
4Ø2Ø FOR N=Ø TO 7: PRINT AT
      8+N,21;"         ": NEXT N
4Ø25 PRINT AT 6,22; PAPER 7;
      INK 2; FLASH 1;
      "SOLUCION"
4Ø3Ø INK 4: PAPER 7: PRINT AT
      9,22;"4333337";AT 15,
      22;"1333332": FOR
      N=1Ø TO 14: PRINT AT N,
      22;"5 5": NEXT N
4Ø5Ø FOR N=51 TO 99 STEP
      16: PLOT 18Ø,N: DRAW
      47,Ø: DRAW Ø,1: DRAW
```

```
      -47,Ø: NEXT N
4Ø6Ø FOR N=179 TO 227 STEP
      16: PLOT N,52: DRAW Ø,
      47: DRAW 1,Ø: DRAW Ø,
      -47: NEXT N
4Ø7Ø FOR N=Ø TO 7 STEP 2:
      PLOT N+177,N+1Ø5:
      DRAW 56,Ø: DRAW Ø,-56:
      NEXT N
41ØØ FOR N=1 TO 9: PRINT AT
      1Ø+INT ((N-1)/3)*2,
      21+(N-INT ((N-1)/
      3)*3)*2; INK 1;S(N): NEXT N
412Ø GO TO 353Ø
5ØØØ REM C.M.PARA EL BORDE
5Ø2Ø RESTORE 51ØØ
5Ø3Ø FOR N=63ØØ1 TO 63Ø17:
      READ W: POKE N,W: NEXT N
5Ø4Ø RETURN
51ØØ DATA 14,255,6,255,12Ø,
      211,254,16,251,13,32,
      246,62,8,211,254,2Ø1
955Ø REM C.M.PARA LETRAS
956Ø RESTORE 958Ø
957Ø FOR N=65ØØØ TO 65Ø11:
      READ U: POKE N,U: NEXT
      N
9575 RETURN
958Ø DATA 33,Ø,61,17,Ø,25Ø,
      1,Ø,3,237,176,2Ø1
96ØØ DATA Ø,126,98,114,1Ø6,
      1Ø2,126,Ø
96Ø5 DATA Ø,56,24,Y,Y,Y,126,Ø
96Ø6 DATA Ø,126,6,126,96,Y,
      126,Ø
9615 DATA Ø,126,6,126,6,Y,
      126,Ø
962Ø DATA Ø,14,22,38,7Ø,126,
      6,Ø
9625 DATA Ø,126,96,Y,126,6,
      126,Ø
963Ø DATA Ø,126,96,126,98,Y,
      126,Ø
9635 DATA Ø,126,6,6,12,24,
      48,Ø
964Ø DATA Ø,126,98,6Ø,98,Y,
      126,Ø
9645 DATA Ø,126,7Ø,126,6,7Ø,
      126,Ø
965Ø DATA Ø,126,98,126,98,Y,
      Y,Ø
9655 DATA Ø,126,98,124,98,Y,
      126,Ø
966Ø DATA Ø,126,96,Y,Y,Y,126,
      Ø
9665 DATA Ø,126,98,Y,Y,Y,126,
      Ø
967Ø DATA Ø,126,96,126,96,Y,
      126,Ø
9675 DATA Ø,126,96,126,96,Y,
      Y,Ø
968Ø DATA Ø,126,96,Y,11Ø,98,
      126,Ø
9685 DATA Ø,98,Y,126,98,Y,Y,Ø
969Ø DATA Ø,24,Y,Y,Y,Y,Y,Ø
9695 DATA Ø,6,Y,Y,Y,7Ø,126,Ø
97ØØ DATA Ø,1ØØ,Y,126,98,Y,Y,
      Ø
97Ø5 DATA Ø,96,Y,Y,Y,98,126,Ø
971Ø DATA Ø,126,1Ø6,Y,Y,Y,Y,Ø
9715 DATA Ø,122,1Ø6,Y,Y,Y,
      11Ø,Ø
972Ø DATA Ø,126,7Ø,Y,Y,Y,126,Ø
9725 DATA Ø,126,98,126,96,Y,
      Y,Ø
973Ø DATA Ø,126,98,Y,1Ø6,Y,
      126,Ø
9735 DATA Ø,126,98,124,98,Y,
      Y,Ø
974Ø DATA Ø,126,96,126,6,Y,
      126,Ø
9745 DATA Ø,126,24,Y,Y,Y,Y,Ø
975Ø DATA Ø,98,Y,Y,Y,Y,126,Ø
9755 DATA Ø,98,Y,Y,Y,52,8,Ø
976Ø DATA Ø,1Ø6,Y,Y,Y,Y,126,Ø
9765 DATA Ø,66,1Ø2,24,Y,1Ø2,
      66,Ø
977Ø DATA Ø,98,Y,126,24,Y,Y,Ø
9775 DATA Ø,126,6,12,24,48,
      126,Ø
98ØØ REM LETRAS
98Ø5 POKE 236Ø6,Ø: POKE
      236Ø7,249
981Ø RANDOMIZE USR 65ØØØ
982Ø RESTORE 96ØØ
983Ø FOR X=64128 TO 642Ø7:
      READ Y: POKE X,Y: NEXT X
984Ø FOR X=64264 TO 64471:
      READ Y: POKE X,Y: NEXT X:
      RETURN
```

M.ª Luisa Cuervo Herrero

EL ZOCO DE INPUT

Todo se compra y se vende. Los antiguos zocos fueron lugares destinados a todo tipo de transacciones. INPUT también tiene el suyo. Vuestras operaciones de compra, cambio o venta serán publicadas en esta sección, pero dos son las limitaciones que imponemos:

a) La propuesta tendrá que ver con la microinformática.
b) Nos reservamos el derecho de no publicar aquellos insertos de los que se sospeche un trasfondo lucrativo.

Ahora un ruego. Tratar de resumir al máximo el texto; escribir casi como un telegrama siendo claros y concisos.

HISTOGRAMAS Y TARTAS ESTADISTICAS

Sea cual sea la procedencia de tus datos —un presupuesto familiar, un pequeño negocio, una afición o los mismos indicadores de tu salud—, notarás la diferencia si los representas mediante un diagrama de barras o una tarta redonda.

Ya has podido ver en nuestra revista cómo se escribe un programa que visualiza tus datos en forma gráfica. Otra manera de representar la información numérica sería el diagrama no lineal. Las barras (histogramas) y las «tartas» están entre las representaciones más ampliamente utilizadas de la información comercial y estadística. Y es que ofrecen una ventaja adicional: pueden colorearse y adquirir formas deslumbrantes.

Aparte de esto, cada diagrama se adapta particularmente mejor a un tipo de datos que a otro. Las barras se prestan a ser utilizadas para datos que fluctúan a lo largo de un amplio abanico de valores. El diagrama en forma de tarta está indicado en aquellas ocasiones en que te interese mostrar los distintos valores en proporción a la totalidad: por ejemplo, si tienes que comparar porcentajes.

Los programas que siguen ilustran la manera de instruir a tu ordenador para que te haga ambos tipos de diagramas.

LA RUTINA DE LAS BARRAS

Esta rutina es, en esencia, la misma que la de cualquier gráfico. Una vez reunidos los datos, hay que entrarlos en la memoria, determinar las coordenadas, trazar las barras y añadir los textos. Una de dos: o introduces (INPUT) cada uno de los datos y vas trazando cada barra sobre la marcha, o esperas a tener introducidos todos los datos para empezar la tarea del trazado de barras. Existe una tercera posibilidad: leer (READ) los datos y trazar las barras, cambiándolos siempre que desees dibujar un diagrama diferente. Lo que sí que te aconsejamos es que, sea cual sea el método a seguir, almacenes los datos en una tabla de variables, porque así la máquina identifica fácilmente las coordenadas de cada barra.

LEER LOS DATOS

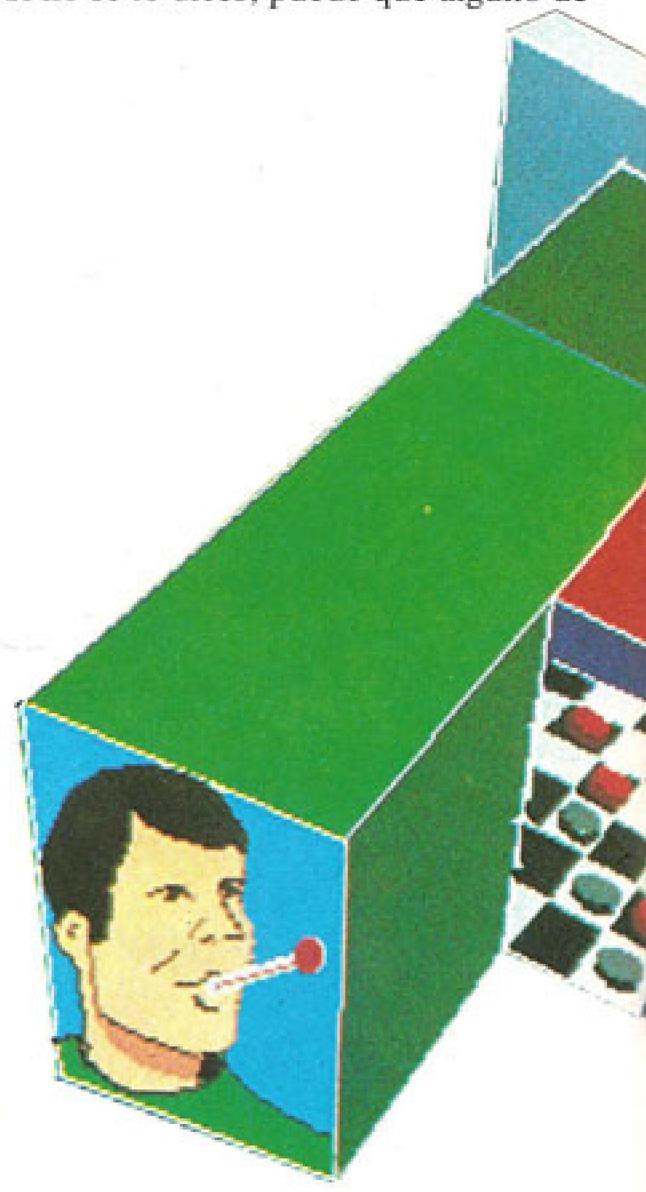
Entra estas líneas que leen (READ) los datos, y ejecútalas (RUN). Todavía no verás nada en pantalla, pero harás bien en ejecutar cada parte del programa, para depurar errores.

```
  1Ø LET n=12
  2Ø DIM a(n)
  7Ø FOR t=1 TO n
  8Ø READ a(t)
  9Ø NEXT t
3Ø1Ø DATA 3,6,5,9,6,3,6,8,3,
     5,9,4
```

Esta sección del programa selecciona un modo que sirva para gráficos en aquellas máquinas que lo requieren. Establece el 12 como número de barras que se trazarán (línea 10) y dimensiona la tabla con este mismo número (línea 20). Escoge, si quieres, otro número, pero si es uno mayor, no olvides que has de tener un número equivalente de entradas en la sentencia DATA de la línea 3010, porque de lo contrario te encontrarás con el error *out of data*. Con frecuencia es mejor disponer de más entradas en la línea de DATA cuando se está comprobando el programa; de este modo puedes aumentar o disminuir el número de barras a trazar en la línea 10, sin tener que alterar los datos cada vez.

LAS ESCALAS

Ya tiene el ordenador en su poder los valores absolutos de cada coordenada de barra, pero habrás de decirle a qué escala los quieres representados. Si no se lo dices, puede que alguno de



los datos te rebase la pantalla, mientras que los datos más pequeños aparezcan en ella como motitas prácticamente invisibles. Para trazar los ejes a escala, escribe y ejecuta (RUN) las siguientes líneas (de nuevo, hazlo como prueba, aunque nada veas todavía en la pantalla).

```
  3Ø LET dx=239/n
  4Ø READ dy
3ØØØ DATA 18
```

Esta parte del programa produce una escala para el eje X dividiendo el área disponible de la pantalla (descontados los márgenes) por el número de

barras que se van a trazar (línea 30). El número máximo de barras posible depende de cada ordenador, pero puedes deducir el valor que mejor gráfico te ofrece si conoces la extensión de la pantalla de gráficos de tu ordenador (la encontrarás en el Manual del Usuario): cambia en consonancia el valor de N en la línea 30.

La escala para los valores del eje Y se obtiene multiplicándolos por un factor, teniendo en cuenta el espacio que necesitas para el texto que acompañará el gráfico. Este valor se entra automáticamente, pero es leído, READ (línea 40), de la sentencia DATA (línea 3000). En este punto, puede que te tiente escribir una breve rutina para entrar, INPUT, este valor en la línea 40. Si sabes cómo hacerlo, borra la línea 3000; de otra manera, éste será el primer tramo de datos que será dibujado, en lugar del primer número contenido en la línea 3010.

El resto del programa comprende dos rutinas, una para dibujar los ejes y otra para dibujar las barras. Escribe esta parte del programa, pero no la ejecutes (RUN), porque recibirás un mensaje de error si el ordenador no encuentra las rutinas que han sido llamadas:

```
100 GO SUB 1000
140 GO SUB 2000
160 STOP
```

Las líneas de la 100 a la 140 llaman las subrutinas que dibujan los ejes y las barras.

APARECEN LAS BARRAS

Entra ahora las rutinas que dibujan los ejes y trazan las barras:

```
1000 PLOT 16,0: DRAW 0,170
1020 PLOT 16,0: DRAW 235,0
1030 RETURN
2000 FOR a=1 TO n
2010 LET s=16+(a-1)*dx
2020 FOR t=s TO s+dx-4
2030 PLOT t,0: DRAW 0,a(a)
     *dy
2040 NEXT t
2100 NEXT a
2110 RETURN
```

Las líneas de la 1000 a la 1020 dibujan los ejes, dejando un margen de 16 unidades gráficas a la izquierda del eje Y. Para dibujar las barras, la línea 2000 establece un bucle, una vuelta por cada barra, la línea 2010 pone escala en la coordenada X (*dx) y la línea 2020 establece un bucle para dibujar las líneas verticales que conforman cada barra. El «−4» al final de esta línea provoca una pequeña separación entre cada barra, para realzar la legibilidad del gráfico. El resto del programa se encarga de dibujar las barras.

TRES DIMENSIONES

Cuando ejecutes este programa, notarás que el diagrama de barras resulta de una apariencia muy poco escolar pero que gana en atractivo a cualquier representación lineal, a pesar de que ambos se basen en los mismos datos. Pues bien, todavía se puede realzar la visualización de la información presentándola en tres dimensiones. Esto posibilita un buen uso del color, y una figura sólida y natural.

Antes de desarrollar el siguiente programa, guarda lo que ya has escrito (para saber lo que tienes) y, sin hacer un NEW o un BREAK, introduce los siguientes cambios. Debes saber evitar las facilidades de tu ordenador para hacer los cambios y ahorrarte esfuerzos, pero sobre todo asegúrate de que no cometes errores dejando pasar pequeñas diferencias en las líneas.

No ejecutes el programa ahora, tal como está, puesto que no está completo y lo único que vas a conseguir es un mensaje de error. Las alteraciones que acabas de hacer establecen algunas variables y los comandos adicionales para dibujar en la dirección Z la tercera dimensión.

Para completar el programa entra a continuación las siguientes líneas y ejecútalo (RUN):

```
1Ø1Ø DRAW 4,4: DRAW Ø,
     −17Ø
2Ø5Ø PLOT 16+(a−1)*dx,
     a(a)*dy
2Ø6Ø DRAW 4,4
2Ø7Ø DRAW dx−
     4,Ø
2Ø75 DRAW −4,−4: DRAW
     4,4
2Ø8Ø DRAW Ø,−a(a)*
     dy
2Ø9Ø DRAW −4,
     −4
```

Ahora ya debes tener sobre tu pantalla una espléndida representación en tres dimensiones de tus datos. Es un útil ejercicio cambiar los valores de los comandos del color para seleccionar aquellos que a ti más te gustan. Anota seguidamente aquellas líneas que habrás de cambiar para alterar la escala del gráfico, de modo que las tengas siempre a mano cuando vayas a utilizar el programa en un futuro. La forma más sencilla de hacer notas es mediante sentencias REM. Por ejemplo, puedes añadir una nueva línea que contenga una sentencia REM seguida de una línea alternativa, que incluye su número y el comando.

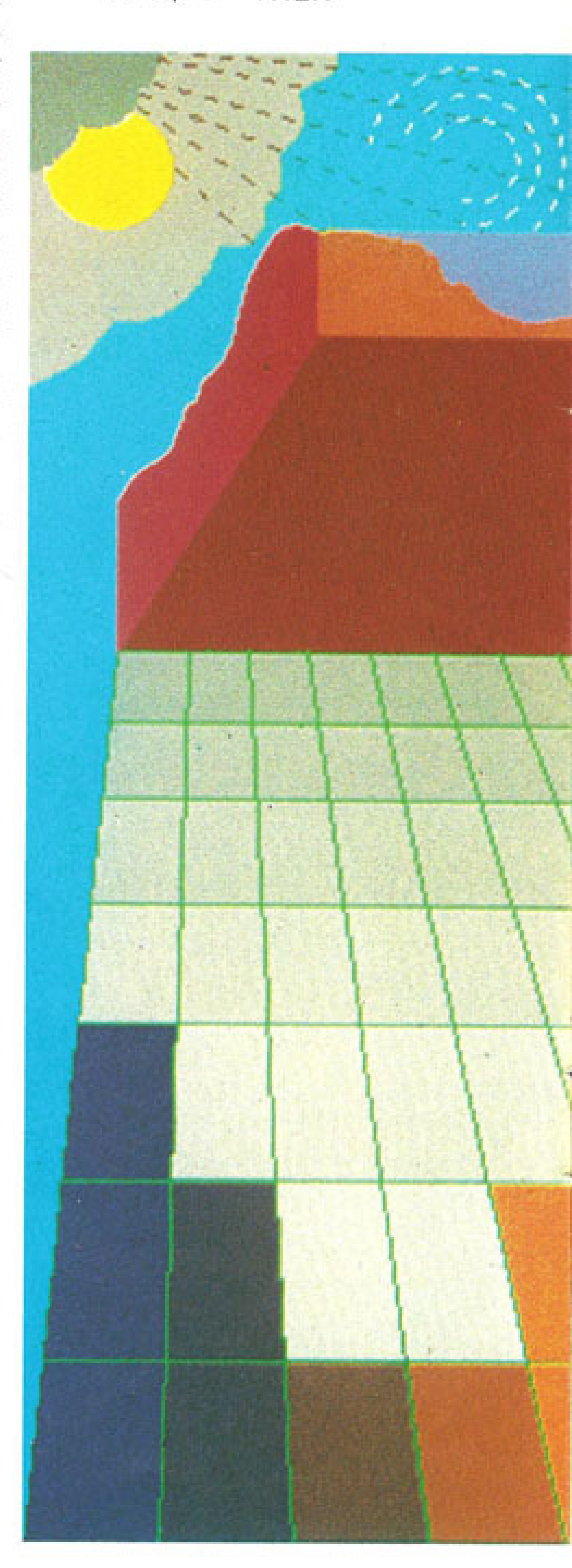
TARTAS

Una manera igualmente atractiva de representar gráficamente la información es el diagrama de tarta. Se trata de un gráfico circular donde se traza una única coordenada (polar) para cada tipo de datos. El tamaño de cada división está representado por un ángulo.

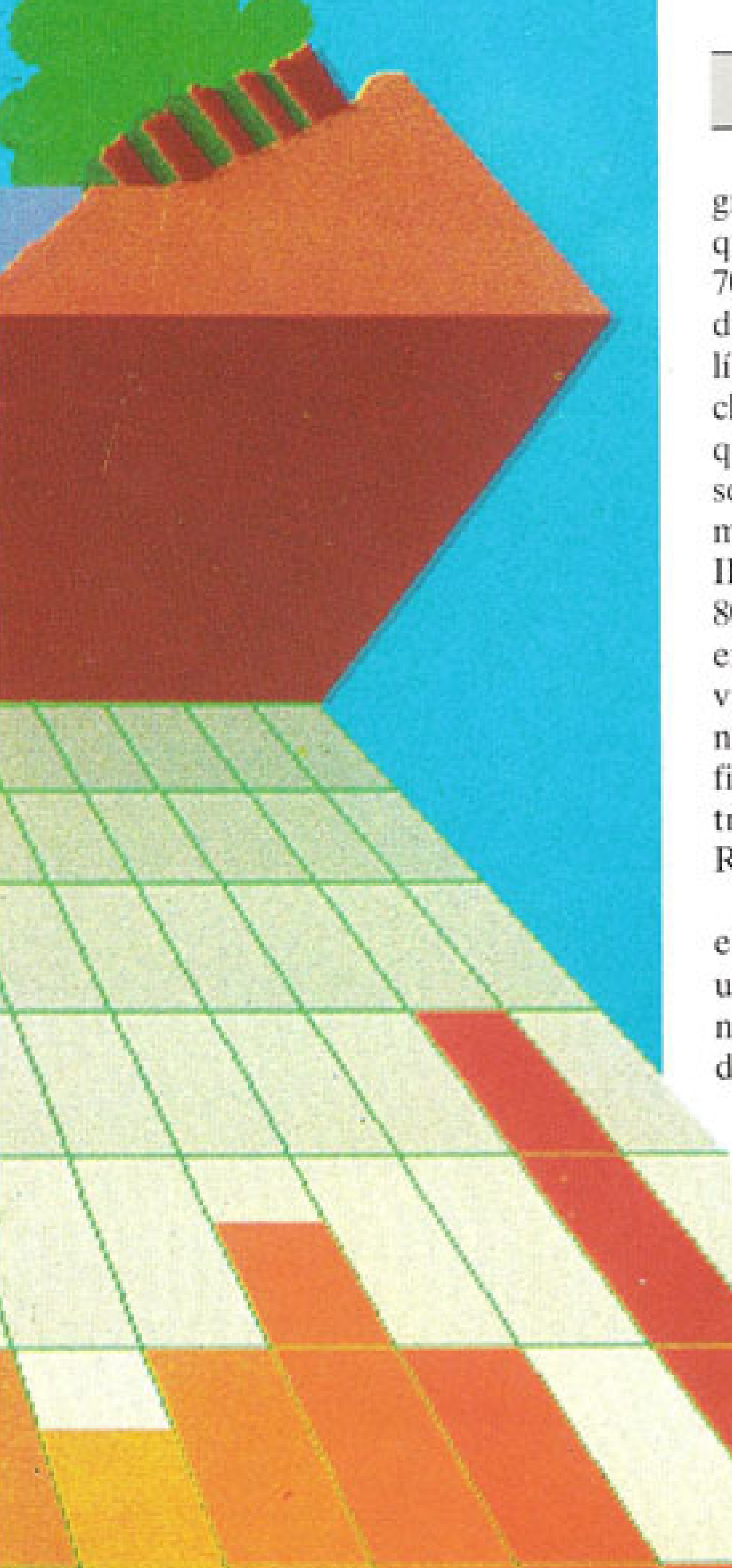
En su estructura, el programa que dibuja una tarta es de lo más sencillo, pero nosotros lo hemos complicado un poco para que lo puedas utilizar con facilidad.

Éntralo y ejecútalo (RUN) si quieres ver cómo funciona:

```
1Ø DIM a(12): LET
   n=Ø
2Ø CLS
4Ø PRINT AT 5,14;
   "MENU"
5Ø PRINT AT 8,1Ø;"1:METER
   DATA"
6Ø PRINT AT 1Ø,1Ø;"2: VER
   CHR$ "
7Ø PRINT AT 12,1Ø;"3:
   FIN"
 8Ø LET a$=INKEY$: IF a$<"1"
    OR a$>"3" THEN GO TO
    8Ø
 9Ø GO SUB VAL a$*
    2ØØ
1ØØ GO TO 2Ø
2ØØ CLS : LET n
    =1
21Ø PRINT "ITEM NO. ";n:
    INPUT LINE a$: PRINT a$:
    IF a$="" THEN
```



```
    RETURN
220 LET a(n)=VAL a$: LET
    n=n+1
230 IF n<13 THEN GO TO
    210
240 LET n=n-1:
    RETURN
400 IF n=0 THEN
    RETURN
410 CLS: LET tt=0: FOR t=1 TO
    n: LET tt=tt+a(t):
    NEXT t
420 LET f=(2*PI)
    /tt
430 CIRCLE 127,86,
    60
440 LET a=0: FOR k=1
    TO n
450 LET m=a+a(k)*f
460 PLOT 127,86: DRAW
    60*SIN m,60*COS m
470 LET a=m
480 NEXT k
490 PAUSE 0: RETURN
```





ANALISIS DEL LISTADO

Cuando hayas ejecutado este programa, verás primero un menú. Es lo que imprime (PRINT) las líneas 40 a 70 y te permite entrar los datos, ver el diagrama o finalizar el programa. La línea 80 espera a que aprietes una tecla; puede ser cualquiera (salvo las que restauran la memoria), aunque sólo 1, 2 y 3 hacen desaparecer el menú. Esto es lo que consigue el IF ... THEN establecido en la línea 80. Una condición semejante situada en la línea 400 hace que el programa vuelva al menú si aprietas 2 sin entrar ningún dato. Y si pruebas a pulsar 3 finalizas el programa. Si quieres que trabaje de nuevo, habrás de recurrir al RUN.

Para empezar, lo más obvio es que elijas 1, que lleva el programa hasta una rutina de entrada de datos. La línea 90 calcula la línea actual a la que debe dirigirse el programa, la cual en este caso es la línea 200. Ésta establece la variable (N) de los datos, que vamos entrando (INPUT) en la línea 210 y almacenándolos mediante la 220 en una tabla que quedó dimensionada en la línea 10.

Para entrar los datos, escribe los números seguidos de ENTER. La línea 230 comprueba si has introducido todos los datos cuyo espacio fue reservado en la línea 10. Si no es así, el programa hace un bucle y vuelve a la línea 210 para entrar el siguiente dato. Observa que no necesitas llenar todo el espacio reservado; si, por ejemplo, sólo deseas una tarta dividida en cinco porciones, habrás de pulsar ENTER dos veces una vez introducido el quinto tipo de datos. La línea 100 lleva entonces el programa a la rutina que visualiza el menú. Ahora bien, si tus datos llenan completamente el espacio reservado, la línea 240 enviará automáticamente el programa al menú, por lo que no podrás nunca rebasar la capacidad de la tabla.

AQUI TIENES LA TARTA

Si ahora pulsas 2, el programa va a la línea 400, que inicia la rutina que dibuja la tarta. La línea 410 hace un bucle para los datos y sus totales. El resto de la rutina determina la escala para éstos y dibuja el gráfico. Repasa, si quieres enterarte bien, lo que ya se ha publicado en INPUT sobre el modo de tratar las funciones circulares de tu ordenador.

La línea 420 divide el gráfico entero (ángulo de 360°, o de 2*PI radianes) por el valor total de los datos para obtener el factor de la escala. La línea 430 dibuja la tarta (o sea, un círculo de radio 60). Las líneas 440 y 450 hacen, entonces, un bucle para los datos entrados, obteniendo el subtotal de éstos en cada pasada y multiplicándolo por el factor de la escala (f). Este subtotal ya a escala (m) es el valor de los puntos de la circunferencia por los cuales se trazarán los radios que dividen el círculo (línea 460).

Y de esta manera ya estás en condiciones de instruir a tu ordenador para que te haga una representación de barras o de tarta.

¡LA MAS GRANDE AVENTURA ESPACIAL DE TODOS LOS TIEMPOS!



LA GUERRA DE LAS GALAXIAS

FREDDY Y LA ARAÑA DE MARTE (y II)

El juego está inicializado. Freddy está esperando en la escalera. Sus flechas son aguzadas, los globos están inflados y a la araña se le hace la boca agua.

En la primera parte de *Freddy y la araña de Marte* introdujiste las rutinas de animación.

Añade estas líneas y tendrás el juego completo:

EL BUCLE PRINCIPAL

```
10 CLEAR 65287
20 GO SUB 1000
30 GO SUB 3000
50 IF ax< >29 THEN GO SUB
   300
70 GO SUB 400
90 GO SUB 500
100 GOSUB 200: IF dead = 0
    THEN GOTO 50
```

El juego está estructurado de modo que se definen los gráficos y se pone a cero la puntuación.

El bucle principal propiamente dicho se extiende desde la línea 50 a la línea 100, continuando mientras Freddy esté vivo. Dicho bucle incluye el movimiento de la flecha, en caso de que se haya disparado, el movimiento de la araña, el movimiento de Freddy de acuerdo con las teclas pulsadas y el movimiento de los globos. Cada una de las rutinas principales actualiza las variables.

LA HORA DE COMER

```
105 LET s(xinc) = 1
110 FOR x=s(xpos) TO 29: GO
    SUB 500: NEXT x
120 LET s(yinc)=1: LET
    s(xinc)=0
125 FOR y=s(ypos) TO 19
130 IF y=my AND ax=29 THEN
    POKE 23607,60: PRINT AT
    my+1,29;" ": POKE 23607,
    252
140 GO SUB 500
150 NEXT y
160 POKE 23607,60: PRINT AT
    10,0; INK 2; PAPER 7;
    BRIGHT 1; "Estas muerto.
    ¿quieres volver a jugar(s-
    n)?";: POKE 23607,
    252
165 LET a$=INKEY$: IF
    a$="" THEN GOTO
    160
170 IF a$="s" OR a$="S"
    THEN GOTO 30
175 IF a$ <> "n" AND a$
    <>"N" THEN GOTO 160
180 POKE 23607,60: CLS:
    STOP
```

Una vez retiradas todas las puertas, «dead» (muerto) se pone a uno y se ejecutan las líneas 105 a 180.

Esta rutina mueve horizontalmente la araña hasta que ésta se encuentra sobre el desventurado Freddy y, a continuación, verticalmente cayendo sobre él y sobre la escalera. Luego se le da al jugador la opción de intentarlo de nuevo. Al colocar el valor 60 en la posición 23607, se restaura el puntero del juego de caracteres, pudiéndose utilizar el juego de caracteres completo.

EL VUELO DE LOS GLOBOS

```
210 LET b(count) = b(count)
    - 1: IF b(count) <> 0
    THEN GOTO 280
220 LET b(count)=b(maxcount):
    PRINT AT b(ypos) + 1,
    b(xpos);" ": LET b(ypos) =
    b(ypos)- 1: IF b(ypos) = 4
    THEN GO SUB 600: POKE
    23607,60: PRINT AT 1,10
    + (3-props)*9;" ";AT 2,10
    + (3-props)*9;" ": POKE
    23607, 252: LET
    props=props-1
225 IF props = 0 THEN LET
    dead=1
230 IF ((ay <> b(ypos) AND ay
    <>b (ypos)+1) OR (ax<
    b(xpos)- 1 OR ax>b(xpos)
    +1)) THEN GOTO 250
240 LET score=score+
    b(points): GO SUB 600: IF
    score>hiscore THEN LET
    hiscore=score: POKE
    23607,60: PRINT AT 0,23;
    INK 0; PAPER 6;hiscore:
    POKE 23607,252
245 GOTO 380
250 GOSUB 4300
280 RETURN
```

b(cuenta) y **b(cuentamax)** son los elementos más importantes de la matriz del globo. Cada vez que se llama la subrutina, la línea 210 decrementa **b(cuenta)**. Cuando dicha variable alcanza el valor cero, el globo se mueve. Una vez movido el globo, la línea 220 copia el valor de **b(cuentamax)** en **b(cuenta)**. El globo puede moverse a distintas velocidades simplemente variando el valor de **b(cuentamax)**. La línea 220 comprueba si el globo ha sido reventado o si ha llegado a la parte superior de la pantalla.

Si el globo ha sido reventado, la puntuación se incrementa. Si ha llegado a la parte superior, se retira una puerta. Si se retiran todas las puertas, **muerto** se pone a uno.

¡TWANG!

```
300 PRINT AT ay,ax;" ": LET
```

```
     ax=ax-1: IF ax<Ø THEN
     LET ax=29: PRINT AT
     my+1,29;"e": LET
     ay=my+1:
     RETURN
31Ø IF ((ay=b(ypos) OR
     ay=b(ypos)+1) AND
     (ax=b(xpos) OR
     ax=b(xpos)+1)) THEN LET
     score=score+b(points): GO
     SUB 6ØØ: IF score>hiscore
     THEN LET hiscore=score:
     POKE 236Ø7,6Ø: PRINT AT
     Ø,23; INK Ø; PAPER 6;
     hiscore: POKE 236Ø7,
     252
33Ø IF ax< >29 THEN GO SUB
     41ØØ
34Ø RETURN
```

Esta es la rutina que anima la flecha. Se borra la imagen anterior y la nueva se imprime en la posición siguiente, determinada por la variable **ax**. Dicha variable se decrementa en la línea 300 y, para evitar que la flecha se imprima fuera de la pantalla, cada vez que la misma llega a ser inferior a cero se pone de nuevo a 29. Cuando **ax** tiene el valor 29, la flecha está de nuevo con Freddy y puede ser lanzada utilizando la barra espaciadora.

Si el valor de **ax** es 29, no puede reventarse ningún globo, por lo que se abandona la subrutina. Si la flecha ha sido disparada, **ax<>29**, la línea 310 comprueba si ésta ha golpeado el globo, en cuyo caso incrementa la puntuación. Si un globo ha sido reventado, se llama la subrutina de la línea 600, la subrutina de estallido del globo.

La línea 330 llama la subrutina que dibuja la flecha en la posición de Freddy, si **ax** no indica que la flecha ya se encuentra en dicho lugar.

MOVIMIENTO POR LA ESCALERA

```
4ØØ LET a$=INKEY$: IF a$=" "
     THEN RETURN
41Ø IF a$="Z" OR a$="z"
     THEN GO TO 45Ø
42Ø IF a$="C" OR a$="c"
     THEN GO TO 44Ø
43Ø IF a$< >" " THEN
     RETURN
432 IF ax< >29 THEN
     RETURN
434 LET ax=28: PRINT AT ay,
     29;" ": RETURN
44Ø IF my=19 THEN
     RETURN
445 PRINT AT my,3Ø; INK 6;
     "k1": LET my=my-1:
     PRINT AT ay,29;" ": IF
     ax=29 THEN LET ay=
     ay-1
47Ø GO SUB 4ØØØ: RETURN
```

Las líneas 400 a 430 leen el teclado. Las líneas 430 y 440 comprueban si se ha pulsado la barra espaciadora y, a continuación, si Freddy dispone de una flecha.

A medida que Freddy se mueve arriba y abajo, se deben usar caracteres de escalera para rellenar el espacio situado encima o debajo de él (ya que si no, desaparecerá la escalera). Si **ax** es igual a 29, la flecha también deberá moverse.

PATAS CORTAS, GRUESAS Y PELUDAS

```
5ØØ LET temp=s(xpos)+(sxinc)
51Ø IF temp<1 OR
     temp>8+(3-props)*9
     THEN LET s(xinc)=s(xinc):
     GO TO 5ØØ
52Ø POKE 236Ø7,6Ø: PRINT AT
     s(ypos),s(xpos);" ";AT
     s(ypos)+1,s(xpos);" ": POKE
     236Ø7,252
53Ø LET
     s(ypos)=s(ypos)+s(yinc):
     LET s(xpos)=temp: LET
     s(picture)=1-s(picture): GO
     SUB 42ØØ
54Ø RETURN
```

La última de las subrutinas de movimiento concierne a la araña marciana. Para hacer más interesante el juego, ésta no se sienta simplemente esperando su comida, sino que pasea arriba y abajo impaciente desde la puerta más próxima hasta el muro del extremo. Hay dos imágenes de araña, almacenándose el número de la imagen actual en **s(imagen)**, la cual se manipula en la línea 530, en donde se obtiene un cero o un uno.

Las líneas 500 y 510 aseguran que la araña no escape de la jaula antes de que todas las puertas hayan sido retiradas.

COMO UN GLOBO SONDA

```
6ØØ PRINT AT b(ypos),b(xpos);
     BRIGHT 1; INK b(colour);
     "gh";AT b(ypos)+1,b(xpos);
     "ij"
61Ø POKE 236Ø7,
     6Ø
62Ø PRINT AT Ø,14; INK Ø;
     PAPER 6;score
63Ø BEEP .5,-20
635 LET bl=bl-1: PRINT AT Ø,
     7; INK Ø; PAPER 6;bl;: IF
     bl=9 THEN PRINT INK Ø;
     PAPER 6;" "
637 IF bl=Ø THEN LET
     bl=15+5*level: LET
     level=level+1: LET
     props=props-1: PRINT INK
     Ø; PAPER 6;AT Ø,7;bl;AT Ø,
     2;level: GO SUB
     6ØØØ
64Ø PRINT AT b(ypos),b(xpos);"
     ";AT b(ypos)+1,b(xpos);
     " "
65Ø PRINT AT ay,ax;" ": LET
     ax=29: LET ay=
     my+1
66Ø POKE 236Ø7,252: GO SUB
     4ØØØ: GO SUB 5ØØØ:
     RETURN
```

Lo único que falta es añadir una rutina que haga estallar el globo si una flecha ha dado en el blanco. La rutina es muy simple, imprimiendo la imagen del globo reventado en la pantalla y borrando después dicha imagen. Si es necesario, se modifica el número de globos que quedan, así como el nivel. La flecha vuelve a la posición de Freddy, preparada para ser disparada contra el siguiente globo.

LA LUNA A TUS PIES

En este formidable juego vas a necesitar de toda la habilidad y sangre fría de que seas capaz para maniobrar el módulo lunar de forma que pueda alunizar perfectamente.

La programación de juegos no tiene por qué generar complicados programas para producir un juego independiente y completo. Aquí te presentamos una versión del célebre programa *Módulo lunar* (*Lunar lander*) que contiene gráficos en alta resolución y un control total sobre el aparato.

El juego es completo y muy variado según lo presentamos, pero tú tienes la posibilidad de adaptarlo a tus preferencias personales. Por ejemplo, puede que te guste añadir una rutina del tipo «¿otro intento?» para evitar volver otra vez al RUN una vez que se ha concluido el descenso. O bien puede que desees alterar los gráficos y los sonidos. Todo depende de ti.

CONTROLES

Deberás utilizar las teclas del cursor 5, 7 y 8.

```
1 BORDER 1: INK 7: PAPER Ø:
  CLS : BRIGHT 1
2Ø FOR N=1 TO 5Ø: PLOT
   RND*255,(RND*135)+4Ø:
   NEXT N
7Ø PLOT Ø,Ø: FOR N=1 TO 16:
   READ GX,GY: DRAW GX,GY:
   NEXT N
8Ø DATA 18,3Ø,18,-15,18,
   -8,18,8,16,2Ø,16,5,13,
   -2Ø,16,-8,18,-4,15,Ø,
   1Ø,1Ø,2Ø,2,5,1Ø,-2Ø,1Ø,
   -1Ø,2Ø,-5,18,2Ø
9Ø PRINT AT Ø,4; INK 6;
   PAPER 2; "FUEL:";AT Ø,18;
   "VELOCIDAD:"
11Ø LET LX=RND*24Ø+1Ø:LET
    LY=16Ø-(15+(RND*1Ø)):
    LET XV=RND*15:LET YV=
    Ø:LET F=246
115 GO SUB 4ØØØ
12Ø GO SUB 1ØØØ: GO SUB
    2ØØØ: GO SUB 3ØØØ
13Ø IF LY>2Ø THEN GO TO 12Ø
135 PAUSE 5Ø
14Ø CLS : IF LX<154 OR
    LX>164 OR ABS YV>4
    THEN GO TO 16Ø
15Ø PRINT " !!FELICIDADES LO
    HAS LOGRADO!!": RESTORE
    5ØØØ: FOR N=1 TO 14:
    READ A,B: BEEP A,B: NEXT
    n: GO TO 17Ø
16Ø PRINT AT 1Ø,7; FLASH 1;
    INK 2; PAPER 7;
    "!!!CRASHED!!!": FOR T=1
    TO 5Ø: BORDER RND*7:
    BEEP .Ø1,RND*5: NEXT T
17Ø PAUSE 4ØØ
18Ø STOP
1ØØØ IF LY<16Ø THEN GO SUB
     4ØØØ
```



PROGRAMACION DE JUEGOS

- UN JUEGO COMPLETO
- HABILIDAD Y DECISION
- GRAFICOS LUNARES
- VELOCIMETRO
- CONTROL DE ATERRIZAJE

- ADAPTAR EL PROGRAMA
- EFECTOS SONOROS
- DESASTRES
- PROGRAMAS CON EXITO
- EL MODULO LUNAR

```
1Ø1Ø LET LX=LX+XV: LET
     LY=LY+YV: IF LY<3ØØ
     THEN BEEP .Ø2,LY/5
1Ø3Ø IF LX<5 THEN LET
     LX=LX+245
1Ø35 IF LX>25Ø THEN LET
     LX=LX-242
1Ø36 IF LY>16Ø THEN RETURN
1Ø37 IF LY<1Ø THEN RETURN
1Ø4Ø GO SUB 4ØØØ
1Ø5Ø RETURN
2ØØØ LET YV=YV-.5 IF F<1
     THEN RETURN
2Ø1Ø IF INKEY$="7" AND f>3
     THEN LET YV=YV+1: LET
     F=F-3: RETURN
2Ø2Ø IF INKEY$="5" THEN LET
     XV=XV-.5: LET F=F-1:
     RETURN
2Ø3Ø IF INKEY$="8" THEN LET
     XV=XV+.5: LET F=F-1:
     RETURN
2Ø4Ø RETURN
3ØØØ PRINT AT Ø,1Ø;" "+STR$
     F+" ";AT Ø,28;" "+STR$
     INT YV+" "
3Ø1Ø RETURN
4ØØØ OVER 1: PLOT LX,LY:
     DRAW -5,-1Ø: DRAW 5,
     2: DRAW 5,-2: DRAW -4,
     1Ø
4Ø1Ø OVER Ø:
     RETURN
5ØØØ DATA .2,4,.2,7,.2,5,.2,
     12,.2,Ø,.2,4,.2,4,.2,5,.6,
     7,.2,12,.2,Ø,.2,4,.2,2,
     .6,Ø
```



DIAGRAMAS DE BARRAS

Para efectuar un análisis instantáneo facilita las cifras y, olvidándote de las matemáticas, observa cómo tu micro las convierte en un diagrama de barras a todo color y con un aspecto totalmente profesional.

Todo el mundo habrá visto referencias de los sistemas informáticos de oficina que muestran elaboradas representaciones de datos tales como cifras de compras e inventarios. Esta clase de información, cantidad de cifras virtualmente incomprensibles, se presenta mejor en forma de gráfico o de tabla, que se comprende más fácilmente de una ojeada. Prepararlo a mano es tedioso, pero precisamente es el tipo de trabajo en el que destacan los ordenadores de oficina, con su capacidad para dar una presentación a todo color en cuestión de segundos.

La presentación de dichos datos no sólo se limita a las máquinas de oficina, sino que también es una de las aplicaciones prácticas a que se suele prestar un ordenador doméstico.

No es probable que el usuario medio de dicho tipo de ordenadores maneje algo parecido a la vasta cantidad de información generada incluso por un negocio pequeño, pero hay muchas cosas que pueden analizarse últimamente en un ordenador. Por ejemplo, ¿se incrementaron los ingresos y los gastos hacia fin de año, o disminuyeron los gastos? ¿Has gastado más en software informático (o en otros artículos tales como revistas, entreteni-

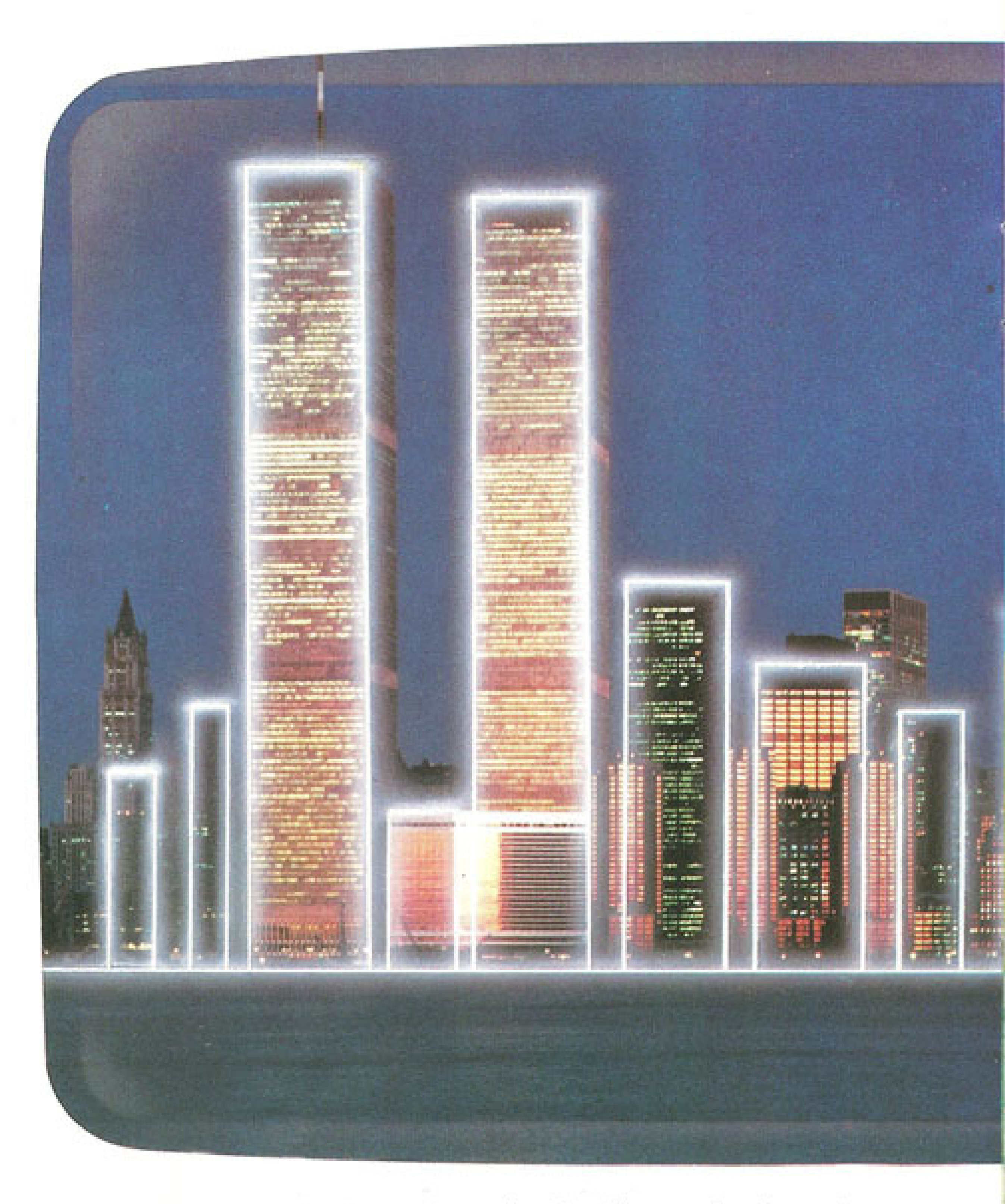


miento y consumo) durante el otoño que durante el verano? ¿Cuándo sobrepasan tus ahorros un cierto nivel y durante cuánto tiempo permanecen así?

Junto a dichas aplicaciones de tipo comercial hay toda clase de temas de interés general, quizá datos y cifras relacionadas con un hobby, que podrían ser analizadas y representadas. Dichos temas van, por ejemplo, desde las cifras de asistencia en el club local hasta los niveles de precipitación o de las mareas. Otras cosas que puedes desear representar gráficamente son resultados deportivos, o el tamaño de una colección en crecimiento.

El sencillo programa ofrecido aquí te permite preparar rápidamente una representación visual de cualquier estadística que varíe a lo largo de un período de tiempo. Los ejes del gráfico se ajustan automáticamente, por lo que puedes entrar cifras semanales, mensuales o anuales. De hecho, puedes utilizar cualquier unidad de tiempo.

Puedes manejar valores que lleguen a cualquier magnitud, unidades, decenas, centenas, millares o incluso millones, si tus cálculos llegan tan lejos.



UTILIZACION DEL PROGRAMA

Al lanzar el programa, visualiza un «menú» o lista de opciones. Seleccionando la opción de introducir nuevos datos, se te invita a indicar los nombres de los ejes. Los rótulos que teclees aparecerán en el gráfico cuando se dibuje el mismo. Cuando introduzcas dichos rótulos, recuerda hacerlo correctamente. El número de barras representando semanas, meses, años o lo que sea, se representa a lo largo del eje x, mientras que el valor de las barras, pesetas o cualquier otra unidad, se representa a lo largo del eje y.

A continuación se te pide que introduzcas el número de barras a trazar. Puedes trazar un máximo de 30 barras.

Luego te pregunta los datos. Imprime el número de cada barra y te pide que introduzcas el valor, que puede ser positivo o negativo. Dichos valores van de −1038 a 1038.

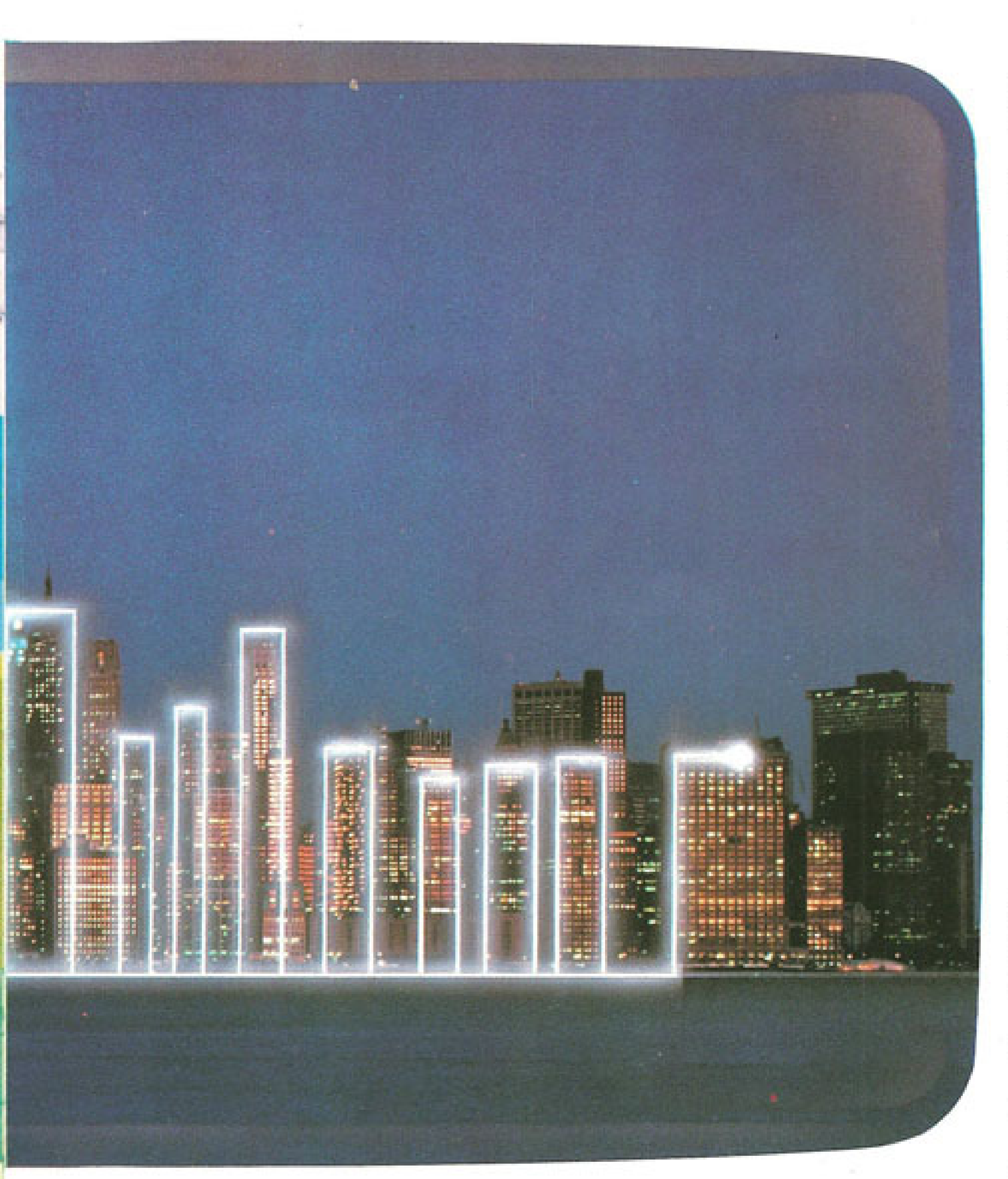
Una vez introducido el último valor, se vuelve al menú de modo que puedas elegir o bien corregir los datos, modificando los valores introducidos en caso de que hayas cometido algún error, o bien trazar el gráfico. Si eliges corregirlos, los valores se imprimirán correlativamente en la pantalla. Debes pulsar cualquier tecla excepto la tecla RETURN para dejar sin cambios el valor y visualizar el siguiente, o pulsar la tecla RETURN e introducir a continuación el nuevo valor.

Cuando estés satisfecho con los valores, selecciona la opción de visualizar o trazar el gráfico.

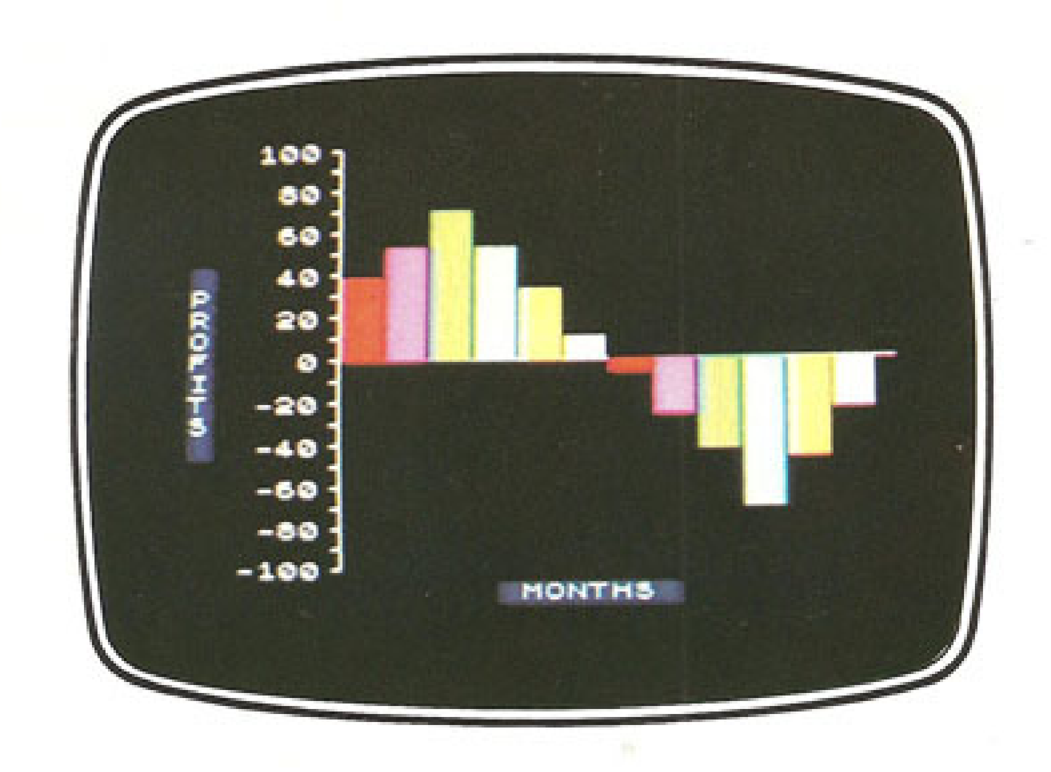
Puedes elegir entre un gráfico escalado o un gráfico a escala completa. La opción de gráfico escalado visualiza el gráfico con las dimensiones correspondientes al eje y redondeadas a un máximo de diez, cien, mil, etc., según el valor máximo de los datos, pero el gráfico no llena la totalidad de la pantalla. La opción de pantalla entera visualiza el gráfico sobre toda el área de la pantalla del televisor, pero imprime los valores reales de los datos (no los valores escalados) a lo largo del eje y.

Para mayor claridad, las barras trazadas van separadas por un pequeño espacio o por una barra coloreada.

```
10 LET 9=0: POKE 23609,20:
   POKE 23658,8
100 BORDER 7: PAPER 7: INK
   0: CLS
```



```
110 PRINT BRIGHT 1; PAPER 3;
    INK 7;AT 4,8;"
    O P C I O N E S"
120 PRINT BRIGHT 1;AT 7,4;"
    1:- ENTRA NUEVOS DATOS
    "
130 PRINT BRIGHT 1;AT 9,4;"
    2:- VER / EDITAR DATOS "
140 PRINT BRIGHT 1;AT 11,4;"
    3:- GRAFICO ESCALONADO
    "
145 PRINT BRIGHT 1;AT 13,4;"
    4:- GRAFICO TODA
    PANTALLA"
150 PRINT BRIGHT 1; FLASH 1;
    INK 2;AT 16,7;"
    SELECCIONA OPCION "
160 IF INKEY$="" THEN GO TO
    160
170 LET a$=INKEY$: IF a$<"1"
    OR a$>"4" THEN GO TO
    160
175 IF a$<>"1" AND 9=0
    THEN GO TO 160
180 GO SUB VAL a$*1000: GO
    TO 100
500 REM ** RUTINA NUMERICA
    DE ENTRADA **
510 INPUT (w$); LINE a$: IF
    LEN a$=0 THEN GO TO
    510
520 FOR j=1 TO LEN a$
540 IF (a$(j)>="0" AND
    a$(j)<="9") OR a$(j)="."
    OR a$(j)="-" THEN NEXT
    j: LET v=VAL a$: RETURN
550 GO TO 510
1000 REM ** RUTINA DE
     ENTRADA **
1010 BORDER 1: PAPER 1: INK
     7: CLS
1020 INPUT "NOMBRE DEL
     EJE-X? "; LINE x$
1030 PRINT INVERSE 1;AT 0,0;
     " " ;x$," "
1040 INPUT "NOMBRE DEL
     EJE-Y? "; LINE y$
1050 PRINT INVERSE 1;AT 0,
     16;" ";y$;" "
1060 LET w$="CUANTOS
     "+x$+" (1 TO 25)? ": GO
     SUB 500
1070 IF v<1 OR v>25 OR
     v<>INT v THEN GO TO
     1060
1090 LET z=v: DIM a(z)
1100 FOR k=1 TO z
1110 LET w$="ENTRA DATOS
     PARA "+STR$ k+" ": GO
     SUB 500
1120 LET a(k)=v
1130 PRINT k,a(k)
1140 NEXT k: LET 9=1: PAUSE
     50: RETURN
2000 REM ** RUTINA DE
     EDITAR/VER **
2010 BORDER 2: PAPER 2: INK
     7
2020 LET cn=1
2025 CLS : PRINT PAPER 6; INK
     2;AT 0,0;x$,y$;TAB 31;" "
2030 PRINT cn,a(cn)
2035 PRINT #1; PAPER 6; INK
     2;AT 0,0;" EDIT para
     cambiar valor actual
     cualquier tecla para
     continuar"
2040 PAUSE 0
2050 IF INKEY$="" THEN GO
     TO 2050
2060 LET c$=INKEY$
2070 IF c$=CHR$ 7 THEN GO
     SUB 2500
2080 IF cn=z THEN PRINT
     PAPER 6; INK 2;"DATOS
     FINALIZADOS": PAUSE
```




```
      100: RETURN
2090  LET cn=cn+1: IF cn=21
      THEN GO TO 2025
2100  GO TO 2030
2500  LET w$="ENTRA NUEVO
      VALOR PARA "+STR$
      cn+" ": GO SUB 500
2510  LET a(cn)=v: PRINT
      PAPER 6; INK 2;cn,a(cn);
      TAB 31;" ": RETURN
3000  REM ** GRAFICA EN
      ESCALA **
3010  BORDER 0: PAPER 0: INK
      7: CLS : LET hi=0: LET
      lo=0
3020  FOR k=1 TO z
3030  IF a(k)>hi THEN LET
      hi=a(k)
3040  IF a(k)<lo THEN LET
      lo=a(k)
3050  NEXT k
3060  LET type=2: LET org=4
3070  IF lo<0 THEN LET type=1:
      LET org=84
3080  LET h=hi: IF ABS lo>hi
      THEN LET h=ABS lo
3090  LET ra=hi-lo
3100  IF h<=1 THEN LET hi=1:
      GO TO 3150
3110  IF h<=10 THEN LET
      hi=10: GO TO 3150
3120  IF h<=100 THEN LET
      hi=100: GO TO 3150
3130  IF h<=1000 THEN LET
      hi=1000: GO TO 3150
3140  IF h<=10000 THEN LET
      hi=10000: GO TO 3150
3150  LET wd=INT (25/z)
3160  IF type=1 THEN LET
      ft=80/hi
3170  IF type=2 THEN LET
      ft=162/hi
3180  PLOT 56,org: DRAW 198,0
3190  PLOT 55,4: DRAW 0,160
3200  FOR n=4 TO 168 STEP 8:
      PLOT 52,n: DRAW 3,0:
      NEXT n
3220  PRINT #1; PAPER 1;AT 0,
      14;" ";x$;" "
3225  LET z$=" "+y$+" "
3230  FOR n=1 TO LEN z$
3240  PRINT AT n+(19-LEN
      y$)/2,0; PAPER 1;z$(n)
3250  NEXT n
3255  LET dc=1
3260  IF type=1 THEN GO TO
      3320
3270  FOR n=hi TO 0 STEP
      -(hi/10)
3275  IF n<.01 THEN LET n=0
3280  LET n$=STR$ n
3290  PRINT AT dc,(6-LEN n$);
      n
3295  LET dc=dc+2
3300  NEXT n
3310  GO TO 3400
3320  FOR n=hi TO -hi STEP
      -(hi/5)
3330  IF n<.01 AND n>-.01
      THEN LET n=0
3340  LET n$=STR$ n
3350  PRINT AT dc,(6-LEN n$);
      n
3360  LET dc=dc+2
3370  NEXT n
3400  LET ink=1
3410  FOR n=1 TO z
3420  LET cm=org
3430  LET ink=ink+1: IF ink=8
      THEN LET ink=2
3440  INK ink
3450  FOR m=1 TO (a(n)*ft)
      STEP SGN a(n)
3460  PLOT 56+(N-1)*wd*8,
      cm: DRAW wd*8-2,0
3470  LET cm=cm+SGN a(n)
3480  NEXT m
3490  NEXT n
3500  IF INKEY$<>"" THEN GO
      TO 3500
3510  IF INKEY$="" THEN GO
      TO 3510
3520  RETURN
4000  REM ** GRAFICO TODA
      PANTALLA **
4010  BORDER 0: PAPER 0: INK
      7: CLS : LET hi=0: LET
      lo=0
4020  FOR n=1 TO z
4030  IF a(n)>hi THEN LET
      hi=a(n)
4040  IF a(n)<lo THEN LET
      lo=a(n)
4050  NEXT n
4060  LET ra=hi-lo: LET
      ft=175/ra: LET
      org=(ra-hi)*ft
4070  LET wd=INT (25/z)
4080  PLOT 56,org: DRAW 198,0
4090  PLOT 55,0: DRAW 0,175
4100  PRINT #1; PAPER 1;AT 0,
      14;" ";x$;" "
4110  LET z$=" "+y$+" "
4120  FOR n=1 TO LEN z$
4130  PRINT AT n+(19-LEN
      y$)/2,0; PAPER 1;z$(n)
4140  NEXT n
4150  PRINT AT 0,0;hi;AT 21,0;
      lo
4200  GO TO 3400
```



EL COLOR EN TU SPECTRUM (II)

En el número anterior publicamos el primer capítulo de una nueva serie dedicada al color, **un aspecto fundamental en la programación de juegos al que generalmente se presta muy poca atención.**

Con el ánimo de sentar los principios que debían servir de base a posteriores explicaciones más complejas, en aquella ocasión nos limitamos a desglosar algunos conceptos básicos, dirigiéndose especialmente a aquellos lectores que todavía no disfrutaban de un buen nivel de programación.

Esta vez volvemos a las páginas de INPUT con el listón un poco más alto, y siguiendo la pauta ascendente que prometimos en el primer capítulo, hablaremos del acceso directo a los **códigos de control** y de la variable **BORDER**, entre otros temas.

EL COMANDO BORDER

El comando **BORDER** es prácticamente un desconocido para muchos aficionados a la programación. Efectivamente, todos sabemos para qué sirve y cómo se usa (y si alguno no lo sabe que corra a por el manual), pero generalmente nos limitamos a emplearlo como una prolongación del comando PAPER. La razón que explica este comportamiento no puede ser más evidente: usando la estrecha terminología de algunos tratados poco recomendables, podríamos decir que BORDER es la única orden referida a la zona *inútil* de la pantalla. De hecho, en la mayoría de las ocasiones se usa únicamente para enmarcar alguna presentación poco vistosa, o para adecuar el color del contorno al tono elegido en el resto de la pantalla.

Sin embargo, con un poco de habilidad e imaginación, no es muy difícil conseguir que esta zona *inútil* participe también en la comunicación con el usuario, a través del único comando que opera en ella.

Veámoslo con un ejemplo:

Si pretendemos enfatizar, en un juego de aventuras, los impactos que recibe el protagonista cada vez que se ve sometido al fuego enemigo, es más que probable que no baste con emitir una nota sincopada a través del débil altavoz del Spectrum. Tampoco sería muy eficaz señalar con un oportuno flash en el *Score* que se han perdido mil puntos, como se hacía en los viejos tiempos. Pero como casi todo tiene arreglo en esta vida, siempre podremos echar mano al comando **BORDER** para ambientar las explosiones con efectistas parpadeos en el contorno de la pantalla, simulando las interferencias que la onda expansiva crearía en el monitor.

Prueba el siguiente programa:

```
10 PAPER 0: INK 7: BORDER 0
20 CLS
30 FOR N = 1 TO 50
40 BORDER 2: BORDER 3:
   BORDER 4:
   BORDER 5: BORDER 6
50 NEXT N
```

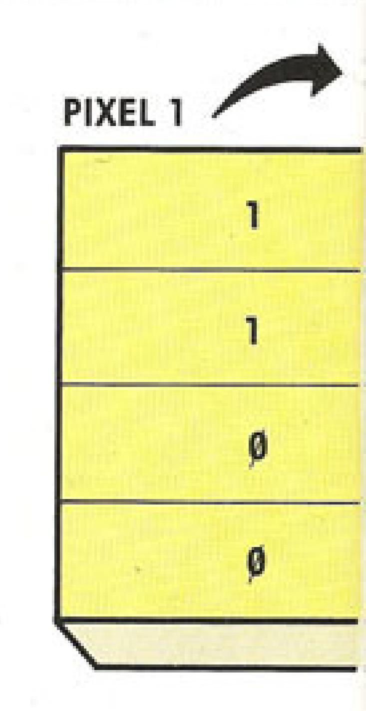
Introduciendo algunas variantes, podemos obtener un resultado menos violento, adecuado a otros menesteres más pacíficos, como por ejemplo un soberbio lucimiento en una pantalla de presentación.

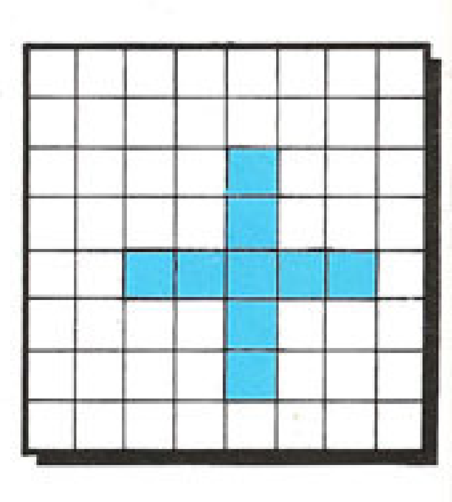
De este modo:

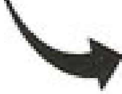
```
10 PAPER 0: INK 7: BORDER 0
20 CLS
30 BORDER 1: BORDER 2:
   BORDER 3:
   BORDER 4: BORDER 5:
   BORDER 6:
   BORDER 7:
   BORDER 0
40 GOTO 30
```

(para detenerlo, usa BREAK, ya que es un bucle infinito)

Como habrás podido apreciar, el programa funciona simplemente re-







corriendo una y otra vez los ocho colores básicos, creando así un efecto *arcoiris*.

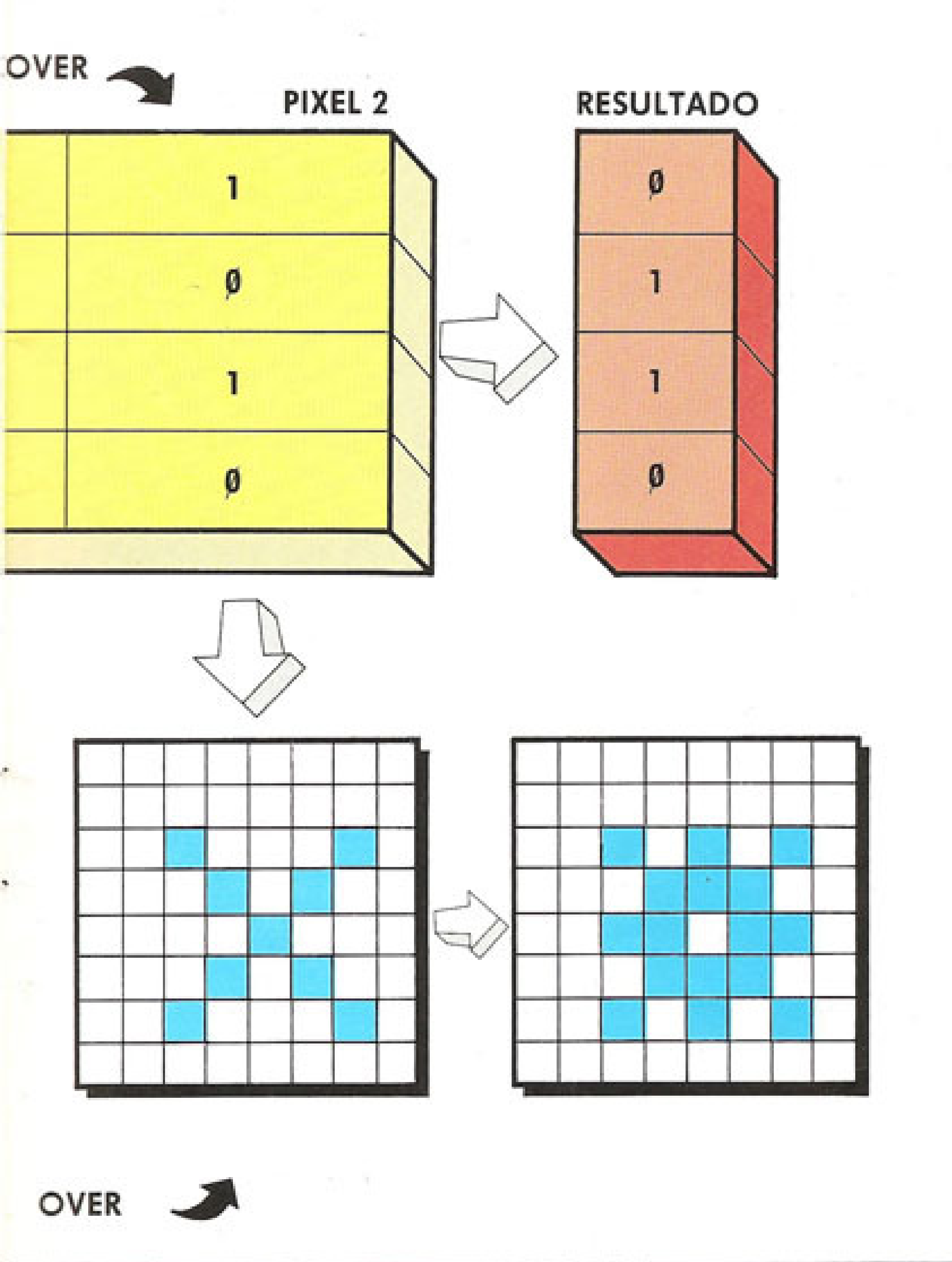
Ahora, añade al programa 2 la siguiente línea y agárrate!!

```
35 PAUSE 1
```

Aunque en principio sorprenda, el resultado obtenido en los dos programas anteriores puede acabar aburriendo. Para evitarlo, debes buscar un efecto más aleatorio y menos repetitivo usando la función **RND**, y operando con ella de forma que obtengas un número al azar entre cero y siete, como mostramos en el siguiente programa:

```
10 PAPER 0: INK 7: BORDER 0
20 CLS
30 LET N = INT (RND * 7):
   BORDER N
40 GOTO 30
```

Variando el número de comandos BORDER en los anteriores ejemplos (especialmente en los programas uno y dos), o introduciendo en el lugar oportuno una sentencia PAUSE, es posible variar la velocidad con que aparecen y se mueven los colores en el borde de la pantalla, o incluso detenerlos, como ocurría en el caso anterior al teclear PAUSE 1. El efecto es sin duda insólito: el contorno aparece decorado con ocho colores aparentemente inmóviles, cuando se supone que no puede haber simultáneamente más de uno.



Si ya has probado el efecto mencionado (introduciendo la línea "35 PAUSE 1" en el programa 2), entonces teclea lo siguiente para ilustrar lo que acabamos de decir:

```
10 PAPER 0: INK 7: BORDER 0
20 CLS
30 BORDER 2: BORDER 3:
   BORDER 0
40 GOTO 30
```

Estos efectos se producen por la relación que se establece entre la velocidad de ejecución de las órdenes BORDER, y el tiempo que tarda la televisión en hacer un «barrido de pantalla» (1/50 seg.). También debes saber que la sentencia PAUSE 1 produce un retardo equivalente a ese mismo lapso de tiempo.

LA VARIABLE BORDER

La variable del sistema **BORDER** (capítulo 25 del manual), es la encargada de almacenar el último valor asignado al color del borde de la pantalla (y a las dos últimas líneas, números 22 y 23, normalmente reservadas a la salida de mensajes y a la entrada de datos a través del comando INPUT). Se encuentra ubicada en la dirección 23624 de la RAM, y tiene la peculiaridad molesta, no debes olvidarlo, de almacenar **no** el código correspondiente, sino este mismo número multiplicado por ocho.

Cuando *pokeamos* esta variable con un nuevo valor, el color del borde no sufre ninguna alteración en tanto no sea pulsada alguna tecla. No obstante, el efecto sí es inmediato en las dos últimas líneas de la pantalla. Esto podemos comprobarlo en el siguiente programa:

```
10 FOR N = 1 TO 7
20 POKE 23624, N * 8
30 INPUT "PRUEBA"; A$
40 NEXT N
```

El contenido de la línea 30 puede parecer inverosímil, y sin embargo resulta imprescindible para el correcto

funcionamiento del programa. No hay que olvidar que el efecto obtenido al actuar directamente sobre la variable BORDER, no se muestra mientras no se use el teclado, y de ahí la necesidad de introducir una sentencia INPUT que permita hacerlo.

Es difícil emplear con éxito este sistema trabajando desde BASIC, salvo en algunos casos muy específicos, pero por ello no debes pensar que su utilidad sea nula. Al menos, conviene que sepas que existe la variable BORDER y cómo se usa.

USO DEL COMANDO OVER

Para no salirnos demasiado del tema, pasaremos por alto los detalles concretos del funcionamiento de este comando, y nos centraremos en sus posibles aplicaciones en relación al color. Por el momento, nos limitaremos a señalar que sigue el modelo lógico del condicional OR, cuyo efecto podría describirse así:

«A o B, pero no ambos»

El cuadro de la figura uno lo ilustra con más detalle.

El comando **OVER** es un instrumento fundamental en el desarrollo de programas de juegos. Gracias a él podemos sobreescribir un gráfico sobre otro sin borrar ninguno de los dos, recuperando ambos más tarde tal y como estaban antes de superponerse.

Supongamos que queremos desplazar un cursor sobre un mapa. Normalmente, el cursor borraría a su paso todos los gráficos. Esto podría solucionarse fácilmente usando OVER, pero si el cursor tiene un color diferente al de los caracteres sobre los que discurre, entonces probablemente persistirían los problemas.

El uso de OVER en casos como éste requiere un cuidadoso estudio de todos los detalles, y una cierta soltura en el manejo del color. La solución que nosotros proponemos, además de ser de aplicación en cualquier programa de juegos, puede ser un útil instrumento que te permita conocer en profundidad los secretos de este versátil comando.

```
10 REM DESP. CUR. OVER
20 BORDER 0: INK 5: CLS
30 FOR N = 1 TO 704
40 LET R = 128+INT (RND * 5)
50 PRINT CHR$ R;
60 NEXT N
70 LET X = 10: LET Y = 10:
   GOTO 140
100 PAUSE 0: LET I$ = INKEY$
110 PRINT FLASH 0; INK 5;
    OVER 1;
    AT X, Y; CHR$ 143
120 LET X = X+(1 AND I$="6"
    AND
    X<21) - (1 AND I$="7"
    AND X>0)
130 LET Y = Y+(1 AND I$="8"
    AND
    Y<31) - (1 AND I$="5"
    AND Y>O)
140 PRINT FLASH 1; INK 2;
    OVER 1;
    AT X, Y; CHR$ 143: GOTO
    100
```

El programa desplaza un cursor por la pantalla (usando las teclas 5, 6, 7 y 8), sobre un fondo de gráficos aleatorios, sin borrarlos ni alterarlos. Está realizado en un BASIC un poco avanzado, con sentencias lógicas que quizás no comprendas muy bien. No te preocupes. En todo caso, lo importante por ahora es que te familiarices con ellas.

CÓDIGO	MODO E	MODO E + CAPS SHIFT
0	PAPEL NEGRO	TINTA NEGRA
1	PAPEL AZUL	TINTA AZUL
2	PAPEL ROJO	TINTA ROJA
3	PAPEL MAGENTA	TINTA MAGENTA
4	PAPEL VERDE	TINTA VERDE
5	PAPEL CYAN	TINTA CYAN
6	PAPEL AMARILLO	TINTA AMARILLA
7	PAPEL BLANCO	TINTA BLANCA
8	SIN BRILLO	SIN FLASH
9	CON BRILLO	CON FLASH

ACCESO DIRECTO A LOS CÓDIGOS DE CONTROL

En muchas ocasiones, sobre todo cuando trabajamos con listados muy largos, necesitamos resaltar alguna sección de programa importante, o indicar el comienzo de una rutina de especial interés, con el objeto de facilitar más tarde su localización. El empleo de sentencias **REM** en estos casos no es muy recomendable, ya que pasan desapercibidas con facilidad y además suponen un gasto extra de memoria.

La solución ideal está en cambiar los atributos de las líneas que nos interesan, destacándolos así del resto del listado. El problema está en que no existen comandos BASIC específicos para hacerlo.

Si, por ejemplo, tecleamos

```
INK 2
```

la tinta de todo el listado se tornará roja, sin que podamos determinar hasta dónde queremos que se produzca el cambio de atributos.

Un método adecuado para lograr nuestro propósito es el llamado «acceso directo». Veamos cómo funciona:

Primero introducimos la línea

```
10 REM PRUEBA
```

Después, la editamos a la parte inferior de la pantalla (pulsando Caps Shift + 1), y colocamos el cursor entre el número de línea y la sentencia REM. Por último, ponemos el cursor en modo E (Caps Shift + Symbol Shift), y pulsamos un número del cero al siete. Inmediatamente, el papel de la línea cambiará de color, según el número escogido.

Si ahora introducimos.

```
20 REM OTRA PRUEBA
```

observaremos que el cambio de atributos también afecta a la nueva línea. Ello se debe a que este procedimiento altera el color de todo el listado a partir del punto en que se produjo el cambio. Para comprobarlo, teclea

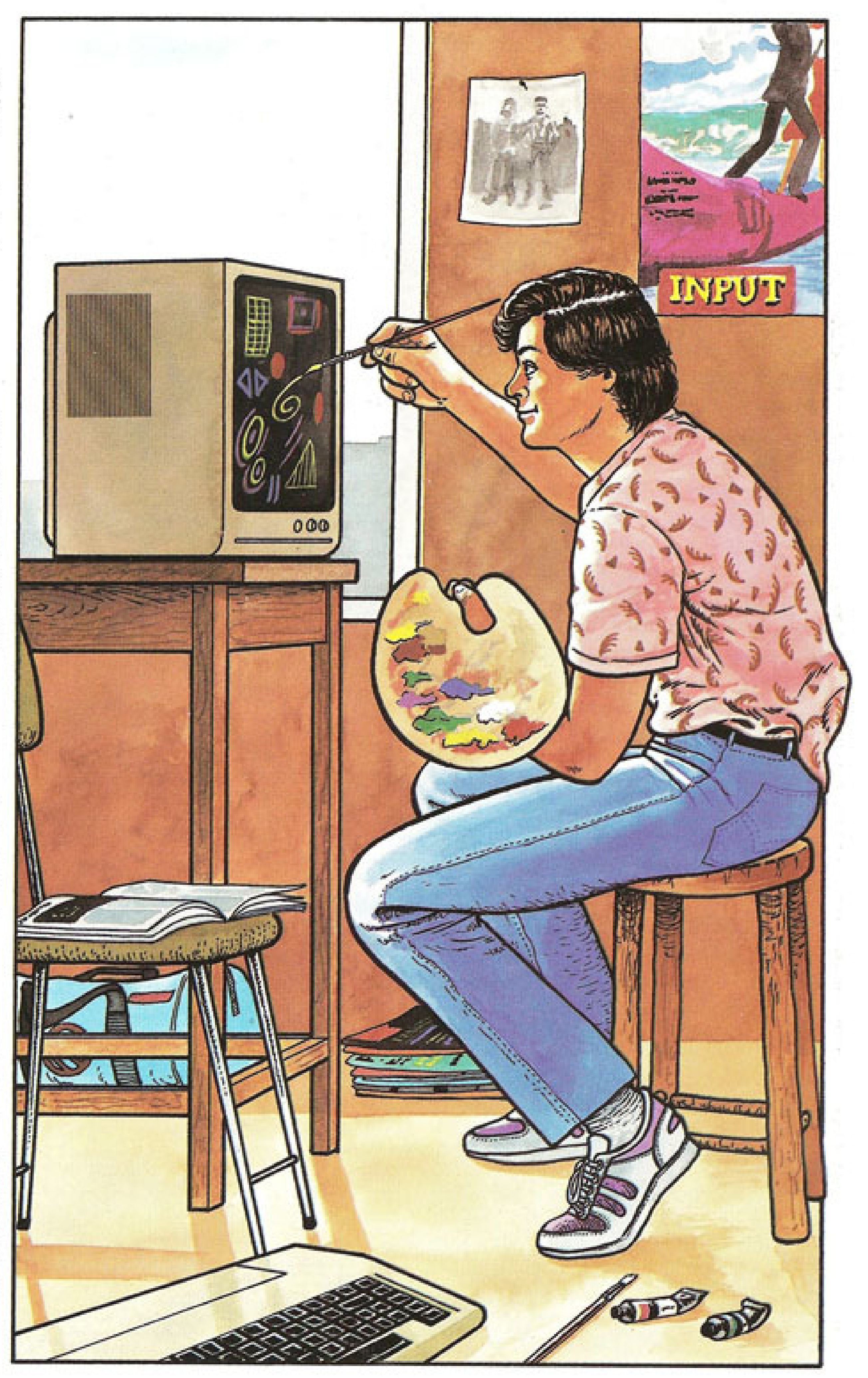
```
5 REM OTRA MAS
```

Como ves, esta nueva línea no se ha visto afectada, por ser anterior a la línea 10, y conserva los atributos originales.

Ahora edita de nuevo la línea 10 y coloca el cursor al final, poniéndolo en modo E. Pulsa el código 7 y seguidamente ENTER. Lo que has hecho esta vez ha sido restituir el color del papel a partir del comienzo de la línea siguiente, es decir, ya tienes una sola línea en un color diferente al resto.

Así pues, el método consiste en alterar los atributos mediante el sistema que hemos descrito, al principio de la línea que nos interesa resaltar, y restituirlos al final. Naturalmente, esto también es aplicable a un conjunto de líneas.

Para cambiar el color de la tinta debes proceder de igual forma, pulsando simultáneamente el código de color elegido y Caps Shift. Por otra parte, el cuadro de la figura dos resume las distintas combinaciones posibles.



GRAFICOS DEFINIDOS POR EL USUARIO (II)

He aquí dos nuevos personajes —una serpiente y un mono— para completar el escenario de la jungla, más algunas ideas para animar el cuadro y para ahorrar memoria.

En el capítulo anterior de esta serie ya has visto la manera de definir una gran cantidad de figuras en tu ordenador y has empezado a formar un cuadro de una escena de la jungla para utilizarlas.

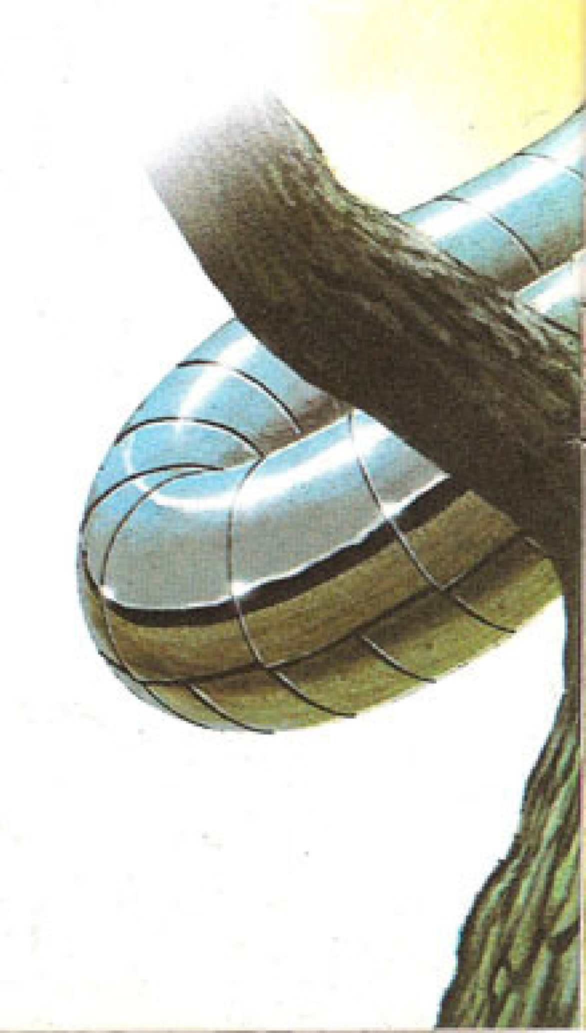
El programa de la escena de la jungla está formado por secciones, en las que se definen en cada una los UDG. Cada sección forma parte del cuadro, el cocodrilo, el elefante, los árboles, etc. La segunda parte del programa que sigue añade una serpiente y un mono y termina el fondo.

Si has almacenado la primera parte del programa en cinta o en otro soporte, podrás cargarlo en el ordenador, porque este programa requiere la primera parte para funcionar.

Las líneas adicionales de datos que necesitas para definir las imágenes se escriben después del programa principal. En primer lugar escribe estas líneas:

```
130 REM Pokear datas de la
     serpiente
140 POKE 23676,254
150 FOR n=USR "a" TO USR
     "r"+7: READ a: POKE n,a:
     NEXT n
160 POKE 23676,253
170 FOR n=USR "a" TO USR
     "j"+7: READ a: POKE n,a:
     NEXT n
180 REM Pokea datas del mono
190 POKE 23676,252
200 FOR n=USR "a" TO USR
     "u"+7: READ a: POKE n,a:
     NEXT n
210 POKE 23676,251
220 FOR n=USR "a" TO USR
     "u"+7: READ a: POKE n,a:
     NEXT n
230 POKE 23676,250
240 FOR n=USR "a" TO USR
     "9"+7: READ a: POKE n,a:
     NEXT n
500 REM Print serpiente
510 INK 1
520 POKE 23676,254
530 LET z=144: FOR n=2 TO 7:
     FOR m=16 TO 18: PRINT
     AT m,n;CHR$ z: LET
     z=z+1: NEXT m: NEXT n
540 POKE 23676,253
550 LET z=144: FOR n=8 TO
     12: FOR m=17 TO 18:
     PRINT AT m,n;CHR$ z: LET
     z=z+1: NEXT m: NEXT n
600 REM Print mono
610 INK 0
620 POKE 23676,252
630 PRINT AT 0,30;CHR$ 144;
     AT 1,24;CHR$ 145;CHR$
     146;AT 1,28;CHR$ 147;
     CHR$ 148
640 PRINT AT 2,23;CHR$ 149;
     CHR$ 150;CHR$ 151;CHR$
     32;CHR$ 152;CHR$ 153
650 PRINT AT 3,23;CHR$ 154;
     CHR$ 155;CHR$ 32;CHR$
     32;CHR$ 156;CHR$ 157;
     CHR$ 158
660 PRINT AT 4,24;CHR$ 159;
     CHR$ 160;CHR$ 32;CHR$
     161;CHR$ 162;CHR$ 163;
     CHR$ 164
670 POKE 23676,251
680 PRINT AT 5,24;: FOR
     n=144 TO 151: PRINT
     CHR$ n;: NEXT n
690 PRINT AT 6,24;CHR$ 152;
     CHR$ 153;CHR$ 153;CHR$
     153;CHR$ 153;CHR$ 154;
     CHR$ 155;CHR$ 156
700 PRINT AT 7,23;CHR$ 157;
     CHR$ 158;CHR$ 159;CHR$
     32;CHR$ 32;CHR$ 160;
     CHR$ 161;CHR$ 162;AT 8,
     22;CHR$ 163;CHR$ 164
710 POKE 23676,250
720 PRINT AT 8,24;CHR$ 144;
     CHR$ 145;CHR$ 32;CHR$
     32;CHR$ 146;CHR$ 147;AT
     9,24;CHR$ 148;CHR$ 149;
     AT 10,24;CHR$ 150
850 REM sol
855 INK 6
860 FOR n=0 TO 2*PI STEP
     .05: PLOT 70,150: DRAW
     SIN n*12,COS n*12: NEXT
     n
870 FOR n=0 TO 2*PI STEP
     PI/4: PLOT 70,150: DRAW
     COS n*20,SIN n*20: NEXT
     n
970 INK 0
```

A continuación escribe estas líneas de datos para definir las formas de los UDG.



- TOQUES FINALES A LA ESCENA DE LA JUNGLA
- UNA SERPIENTE Y UN MONO
- COMPLETANDO EL FONDO

- EMPLEO DE GRAFICOS DE ALTA RESOLUCION
- ANIMACION DE LOS CARACTERES
- ALMACENAMIENTO DE LOS DATOS EN CODIGO MAQUINA

```
14ØØ REM serpiente
141Ø DATA Ø,Ø,Ø,Ø,Ø,Ø,3,14,
     31,31,63,67,65,254,254,
     225,71,63,Ø,Ø,Ø,Ø,Ø,Ø,Ø
142Ø DATA Ø,Ø,Ø,Ø,255,12,28,
     28,28,254,143,1,Ø,4,
     248,2Ø8,224
143Ø DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,
     Ø,Ø,128,24Ø,56
144Ø DATA 52,99,225,225,
     225,96,47,31,14,6,2,1,
     Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,3,5,63,
     127
145Ø DATA 255,255,24Ø,248,
     252,255,255,126,6Ø,28,
     124,247,247,115,31,1,Ø,
     Ø,Ø,Ø
146Ø DATA 255,191,159,143,
     199,194,127,64,128,
     128,64,63
147Ø DATA 48,112,248,249,
     253,255,189,255,Ø,Ø,Ø,
     Ø,192,252,19Ø,63,63,24,
     24Ø
148Ø DATA 9,5,7,5,249,67,
     243,251,255,252,238,
     239,255
149Ø DATA 192,224,2Ø8,223,
     224,129,193,227,243,
     247,255,248,24Ø,224,
     192
15ØØ DATA Ø,Ø,Ø,Ø,255,129,
     195,199,199
```



```
1510 DATA 239,239,255,0,0,
     0,0,0,0,0,0,128,248,31,
     140,158,191,191,255,0,
     0,0,0
1520 DATA 0,0,0,1,15,62,206,
     30,61,121,243,247,247,
     63,15,3,0
1530 DATA 30,127,191,95,93,
     61,250,240,194,252,
     194,192,192,224,224,
     240
1540 REM mono
1550 DATA 1,2,4,8,16,32,64,
     128,0,0,0,3,31,63,127,
     252,0,0,0,224,248,252,
     252
1560 DATA 62,3,15,15,15,15,
     14,14,12
1570 DATA 3,142,240,192,
     128,0,0,0,1,1,3,3,3,3,3,
     3,248,240,224,224,192
1580 DATA 192,192,192,30,
     14,110,76,56,0,0,0,0,0,
     0,0,0,0,0
1590 DATA 1,28,28,56,56,
     112,112,224,224
1600 DATA 3,3,1,1,1,0,0,0,
     192,224,224,240,248,
     248,252,126,3,7,15,31,
     63,62
1610 DATA 126,252,192,192,
     128,0,0,0,3,7,0,0,0,0,0,
     0,128,223
1620 DATA 127,63,31,31,15,
     7,7,3,0,128,128,192,
     192,192,224,240,252,
     126,126,127
1630 DATA 63,63,31,31,15,
     15,7,3,3,131,131,131
1640 DATA 255,255,231,231,
     255,252,172,183,128,
     192,224,224,224,224,
     224,224
1650 DATA 1,1,0,0,0,1,3,7,
     248,255,255,255,255,
     255,255,255,0
1660 DATA 255,255,255,255,
     255,255,255
1670 DATA 15,255,255,255,
     255,255,255,255,199,
     255,255,255,255,255,
     255,255
```



```
1680 DATA 223,239,255,255,
     252,128,0,128,224,240,
     248,248,240,96,0,0,0,0,
     200
1690 DATA 232,248,248,120,
     112
1700 DATA 7,15,15,31,31,31,
     63,63,255,255,255,255,
     255,255,255,255,192,
     224,224
1710 DATA 224,224,224,192,
     192,0,0,1,1,3,7,14,60,
     240,224,192,128,128,0,
     0,0
1720 DATA 0,0,0,0,0,3,15,31,
     63,63,63,63,126,254,
     252,240,191,191,63,63,
     63
1730 DATA 63,63,63,15,7,7,7,
     7,7,7,7,192,131,135,
     159,191,255,254,252
1740 DATA 252,248,240,224,
     192,128,0,0,0,1,3,7,6,6,
     4,0,127,254,240,224
1750 DATA 64,64,0,0,192,0,0,
     0,0,1,1,3,63,31,63,127,
     254,252,248,240
1760 DATA 7,3,0,0,0,0,0,0,
```



```
     24Ø,192,Ø,Ø,Ø,Ø,Ø,Ø,7,7,
     15,31,31,31,23,23,224
177Ø DATA 192,128,Ø,Ø,Ø,Ø,Ø,
     7,6,4,Ø,Ø,Ø,Ø,Ø
```

Todas estas adiciones se adaptan perfectamente a la primera parte del programa, y lo amplían para añadir al mismo más figuras. De hecho, al programa se le añaden más líneas para introducir más datos.

EL CUADRO COMPLETO

Como podrás ver cuando hagas correr el programa, la pantalla queda ahora llena de objetos y personajes, con lo que el cuadro parece completo. La serpiente, el mono y el sol son añadidos por las líneas adicionales.

También puedes añadir más elefantes, árboles o cocodrilos si lo deseas, pero, en cualquier caso, dispones de un juego completo de personajes para utilizarlos en tus propios cuadros.

UN SOL DE ALTA RESOLUCION

Observa que con los gráficos de caracteres normales se emplean gráficos de alta resolución. De hecho, no hay ninguna razón para que no puedan combinarse ambos, y la mezcla proporciona unos resultados sorprendentemente buenos. El sol en alta resolución se forma dibujando una serie de radios muy juntos entre sí.

Si lo deseas, puedes introducir modificaciones en el cuadro con otros gráficos de alta resolución. Por ejemplo, puedes añadir más colinas o incluso trazar algunas líneas para hacer que el suelo aparezca rugoso o resquebrajado.

SIGUIENTES PASOS

La gama de variaciones que puedes obtener con los UDG es casi ilimitada. Además de cambiar el número de serpientes, de árboles, de monos, etc., también puedes utilizar los caracteres gráficos para que formen estas imágenes para otras cosas.

Un ejemplo evidente consiste en emplear las copas de los árboles con un color diferente (como puede ser el blanco) para nubes, o como arbustos.

En estos casos deberás tener en cuenta los contrastes de color.

Supón que quieres emplear copas de árboles como arbustos: si colocas los arbustos en las colinas, como éstas son verdes, dichos arbustos no se verán, a menos que desees que los arbustos sean rojos, por supuesto.

ANIMACION DEL CUADRO

Las ventajas de crear imágenes con UDG no terminan con las posibilidades de modificar la escena una vez terminada, sino que, con muy pocos cambios, puedes convertirla en un cuadro animado.



Por ejemplo, para animar los animales, puedes definir uno o más caracteres adicionales: otra trompa para el elefante, una serpiente con la cabeza levantada o el mono comiéndose un plátano son sólo algunas de las muchas posibilidades.

Empleando una serie de UDG distintos para las diferentes posiciones y presentándolos sucesivamente, el movimiento se podrá hacer mucho más real, más divertido, más amenazador, o como desees.

COMO SE ENCAJAN LOS UDG

Si vas a emplear la animación con UDG o a crear imágenes por ti mismo podrás encontrar en estas páginas los dibujos de la serpiente y del mono, y el del cocodrilo en el capítulo anterior.

Al final se indican algunas posibles sustituciones de datos para el elefante, que es posible que desees emplearlas para la animación.

Mientras los animales parecen quedar limitados a escenas de jungla o de zoológico, los árboles, las nubes y los arbustos podrás utilizarlos en casi todos los cuadros que desees realizar con un programa. Con métodos similares podrás dibujar por ti mismo las escenas que desees.

AHORRANDO MEMORIA

El programa descrito emplea una gran cantidad de UDG para demostrarte la forma de utilizar tu ordenador más allá de sus límites, así como proporcionarte algunos personajes y figuras que puedes emplear en tus propios cuadros.

Pero si deseas o necesitas crear un cuadro impresionante con bastantes menos UDG, puedes utilizar el mismo UDG una y otra vez. Por ejemplo, puedes crear una manada entera de elefantes. O si en tu cuadro has incluido una pared, podrás cubrir una gran parte de la pantalla con sólo dos UDG.

Planificando cuidadosamente lo que deseas dibujar, podrás crear con mucho éxito cuadros muy interesantes con un número de UDG sorprendentemente reducido. Sin embargo, como es bastante normal que al cabo de un tiempo dispongas de un gran número de UDG, las únicas ventajas que pueden obtenerse *economizando* los UDG son el tiempo que ahorras al no tenerlos que teclear y en la memoria ocupada por los datos. Como el programa emplea una gran cantidad de memoria para almacenar los datos, puedes ahorrar mucha de esta memoria almacenando los bytes que forman los UDG en forma de bloques de memoria y borrando después las sentencias de los datos. Normalmente, el ordenador almacena cada byte dos veces: una en las posiciones de memoria que llenas con los datos (POKE) y también con las propias sentencias de datos.

Para almacenar los datos como bytes de código, emplea:

```
SAVE "nombre fichero" CODE x,y
```

donde x es la dirección de inicio del bloque de memoria que deseas almacenar, e y es la longitud del bloque.

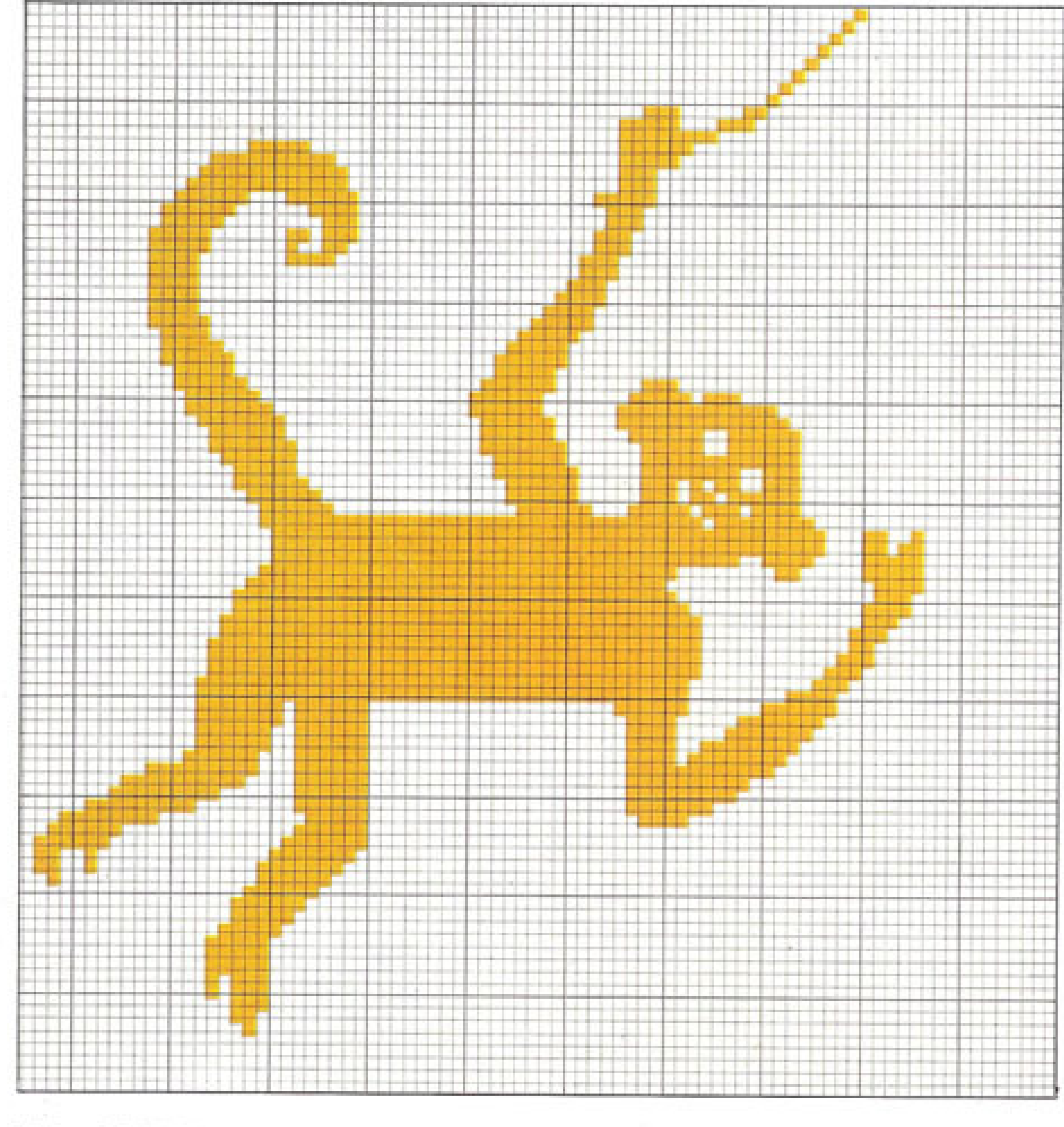
Para ello deberás conocer la dirección de inicio del bloque de memoria, puesto que la necesitas para introducir los datos o para cargar el puntero con la nueva dirección de inicio.

La longitud del bloque es muy sencilla: normalmente es el número de UDG que tienes en la memoria multiplicado por 8. Efectivamente, el anterior programa que define los UDG para el cuadro de la jungla emplea ocho bancos de UDG y empieza por separado cada banco de 256 bytes en lugar de los 158 bytes mínimos. La razón de esto es que te permite variar el puntero con sólo un POKE, en contraposición con los dos habituales. También significa que si quieres almacenar los bytes como códigos, deberás almacenar un bloque de 256* 8 bytes de longitud.

Por tanto, este comando almacena los bytes de los UDG de la jungla:

```
SAVE "UDG de jungla" CODE
63488,2Ø48
```

Para volver a cargar en memoria los bytes sólo hay que teclear LOAD"" CODE. Para volverlo a cargar en una dirección distinta a la que estaba almacenada, te bastará con añadir la nueva dirección de inicio después de LOAD"" CODE.



ANIMA TUS FIGURAS

A continuación se indican algunos datos que puedes incorporar al programa principal para animar el elefante. Simplemente proporciona una alternativa de la posición de la trompa, por lo que puedes pasar de una a otra para simular la animación.

```
DATA Ø,Ø,Ø,Ø,Ø,1,3,3
DATA 6,15,15,31,29,25,26,24
DATA 24,24,12,12,6,Ø,Ø,Ø
```

Además, tendrás que variar los correspondientes bucles FOR ... NEXT para establecer y presentar los nuevos UDG.

También puedes mejorar el elefante animado desplazándolo a una posición más visible de la que aparece en el cuadro.

La animación tiende a realzar un personaje, por lo que es una buena práctica emplear las partes animadas de tu cuadro como la parte principal. De esta manera podrás distraer la atención del observador de las partes menos llenas del cuadro.

MAPA DE SPIRITS

Si el mes pasado os hacíamos entrega del cargador del programa SPIRITS de TOPOSOFT, este mes os ofrecemos su sensacional mapa con las más de 80 pantallas que lo forman.

En la carrera por los programas de laberintos y arcades iniciados en el MSX por el programa SORCERY, ahora TOPOSOFT, recogiendo el guante, ha sabido crear un programa donde han sabido incorporar las mejores ideas de los programas anteriormente hechos, pero sin olvidar de poner nuevos elementos que hicieran de éste un juego nuevo y distinto.

Como comentábamos en el número anterior de INPUT MSX, en el que otorgamos a esta video-aventura nada menos que una calificación total de 42 puntos, lo que en realidad aporta interés y relevancia a la apasionante tarea de innovar el software son los cambios que afectan directamente al planteamiento de la acción del programa, o al desarrollo en la pantalla de la idea original del guionista.

Y este objetivo lo han logrado plenamente los creadores de esta interesante y terrorífica video-aventura, cuyo mapa completo comentaremos ampliamente en esta oportunidad.

Basado en la interminable lucha entre el bien y el mal, que esta vez se desarrollará en un infernal y maléfico castillo, donde nosotros, que representamos a las fuerzas del bien, deberemos proteger a dos personajes que desde nuestra infancia nos fueron fieles compañeros de múltiples aventuras. Pero no lloréis ante tantos recuerdos, y luchad hasta que las fuerzas del mal os obliguen a doblegaros ante ellas (cosa que será muy difícil, casi imposible, si usáis el cargador que se encuentra en el número 13 de INPUT MSX).

Gracias al mapa podrás hacer un mejor uso de elementos mágicos como es la bola de cristal, que te indica dónde se encuentran los restantes personajes del juego. Con el mapa podrás averiguar verdaderamente donde se halla el personaje que estés interesado en controlar, y encontrar el mejor camino para ir hasta él. El mapa también te será de gran ayuda para localizar las diferentes palancas que te sirven para abrir puertas y pasadizos secretos. Pero no todo te será tan fácil como ahora te pueda parecer: en el mapa no aparecen los enemigos a quienes tendrás que enfrentarte, pero recuerda que posees un rayo paralizador que te ayudará a librarte de sus ataques. En caso de ser atacado deshazte rápidamente de tus enemigos, ya que



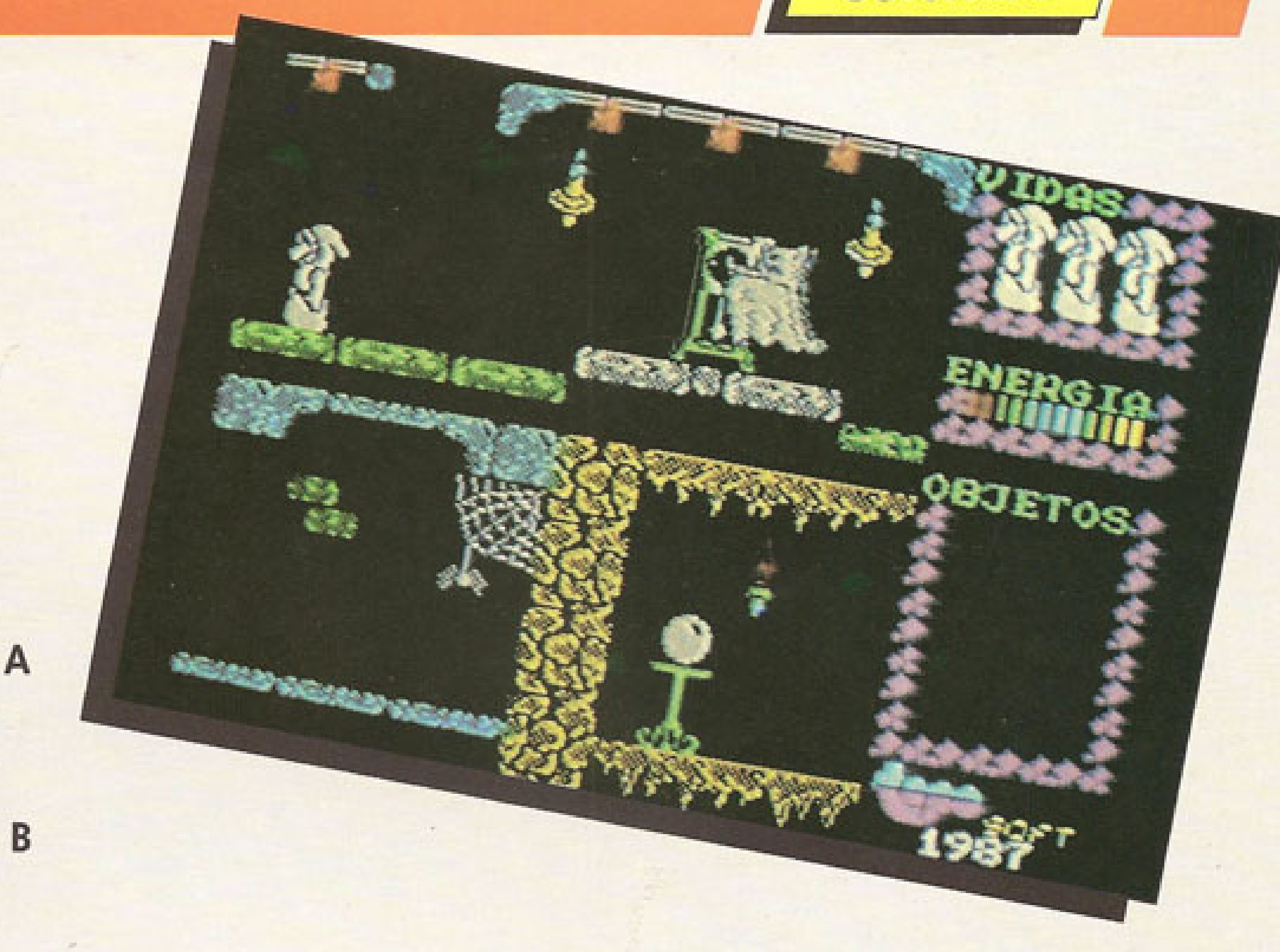
ellos te irán restando energía, y cuando ésta llegue a cero perderás una vida, aunque si has usado el cargador...
La aparición simultánea de cantidad de información en pantalla quizá te despiste un poco, pero en todo momento recuerda que lo más importante es mantener a salvo al mago que tú conduces por el castillo. En el mapa podrás observar la disposición de todas y cada una de las 87 pantallas que lo forman, y gozar de los espectaculares gráficos que adornan este juego. En el castillo encontrarás de todo: escudos, antiguos pergaminos, calderos, etc., pero recuerda que los tres elementos más importantes, y que tú debes conseguir, son la bola de cristal, antes citada, la varita mágica, imprescindible para desencantar a nuestra bella doncella, y el libro de conjuros, que nos servirá para romper el hechizo que liga a la armadura.



A

B

El mapa de la aventura de este mes se llama SPIRITS. Aquí os lo presentamos fotográficamente para que podáis recorrer sus tétricas mazmorras, que rezuman maldad, olvido y terror, con la tranquilidad de llevar una buena guía bajo el brazo: INPUT. No os perdáis por el camino y pensad que el pasadizo A sigue en A y el B continúa en B. Cosa lógica, ¿no?





A

B



SPEED KING: UN PEQUEÑO GRAN JOYSTICK

Son muchos los modelos de joystick disponibles en el mercado para SPECTRUM, y casi todos ellos, salvo alguna excepción (como el CHEETAH 125), se caracterizan por responder a diseños anteriores a 1985. En los últimos años, las únicas novedades aparecidas se han dedicado exclusivamente a cubrir la demanda de los nuevos modelos de la gama SINCLAIR, con sus peculiares condiciones de compatibilidad, olvidando casi por completo a los usuarios del SPECTRUM 48 K. Son muchas y muy variadas las razones que pueden explicar este comportamiento de los fabricantes, pero todas ellas apuntan a la acumulación de stocks, la competencia de los precios, la imposibilidad de incorporar innovaciones sustanciales, y, sobre todo, a un cierto grado de saturación del mercado. Por estos motivos, la reciente aparición del CHEETAH 125 y del polémico ZERO-ZERO han tenido lugar como «rebote» de las versiones para el PLUS 2, y no como diseños originariamente pensados para los modelos de 48 K. Después de lo expuesto, nadie debe extrañarse de que el joystick de más avanzado diseño y de características más novedosas no sea un lanzamiento de 1987, sino un «veterano» que se ha mantenido a la cabeza desde 1985, figurando siempre entre los mejores. Se trata del modelo superanatómico SPEED KING, comercializado por la firma KONIX y relanzado en España por estas fechas.

El diseño del SPEED KING rompe con el tradicional sistema de anclaje con ventosas a una superficie plana, incorporando una base anatómica que se adapta perfectamente a la mano izquierda, mientras la derecha queda libre para maniobrar con la empuñadura (un aspecto negativo: esta ventaja puede convertirse en inconveniente si el usuario es zurdo). Por otra parte, la disposición del botón de disparo hace posible su manejo con la misma mano que sujeta el joystick, sin forzar la posición del dedo, evitando así el cansancio prematuro.

EN EL INTERIOR DE LA PALANCA DE MANDO

En cuanto a las interioridades del SPEED KING, hay que señalar el eje de acero, que garantiza una gran resistencia a los movimientos violentos, y el sofisticado hard de *microswitches*, que proporciona una extraordinaria precisión y rapidez que no ha podido superar ningún otro modelo hasta la fecha.
El único aspecto negativo que podemos reseñar, aparte del ya mencionado, es la inexistencia de un segundo botón de disparo, no imprescindible pero sí muy útil.
Por último, una lista de compatibilidades que muy pocas palancas de mando pueden lucir: SPECTRUM, SPECTRUM PLUS, COMMODORE (64, 128, cl6, plus 4, VIC 20), ATARI, DRAGON, MSX, AMSTRAD y ORIC.
Como podéis apreciar, el joystick comercializado por la firma KONIX puede ostentar una versatilidad realmente extraordinaria.

SPEED KING KONIX
CUADRO-BALANCE

DIMENSIONES: 10 × 13 cm.
LONGITUD DEL CABLE: 165 cm.
DISEÑO: Base y empuñadura anatómicas, sin ventosas.
HARD: Sistema de *microswitches*.
DISPARO: Un solo botón en la base, con disposición anatómica.
VELOCIDAD Y PRECISION: Excelentes.
EJE DE EMPUÑADURA: Acero.
PRECIO ORIENTATIVO: 3.000 ptas.

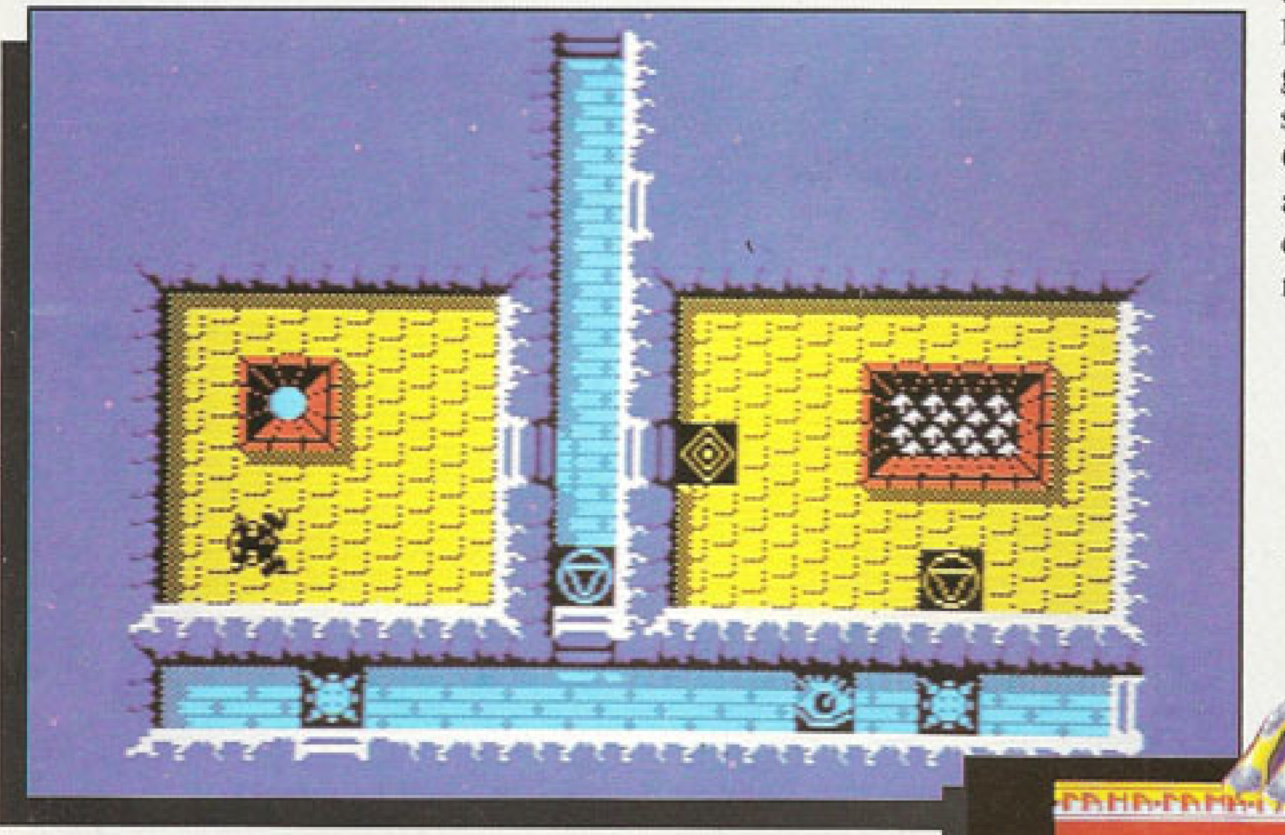
PROGRAMA DEL MES

RANARAMA

● HEWSON ■ ARCADE

La rana Mervyn, en verdad, no era tan rana como parecía a simple vista. Más bien, y hasta cierto punto podía felicitarse por ello: se trataba de un joven aprendiz de mago venido a menos, que se transformó accidentalmente mientras preparaba cierta poción mágica, aunque su propósito, naturalmente, era bien distinto. El pobre Mervyn pretendía convertirse, merced a los buenos oficios alquímicos de un misterioso brebaje, en un musculoso galán de cine, una especie de cruce entre modelo masculino de ropa interior y culturista físico al estilo de Rambo. Desgraciadamente, en la emoción del momento cumbre, confundió uno de los ingredientes, y ahora, en vez del cuerpo apolíneo con el que tanto había soñado, no tenía más que el ridículo aspecto de un insignificante renacuajo. El destino se había ensañado cruelmente con él.

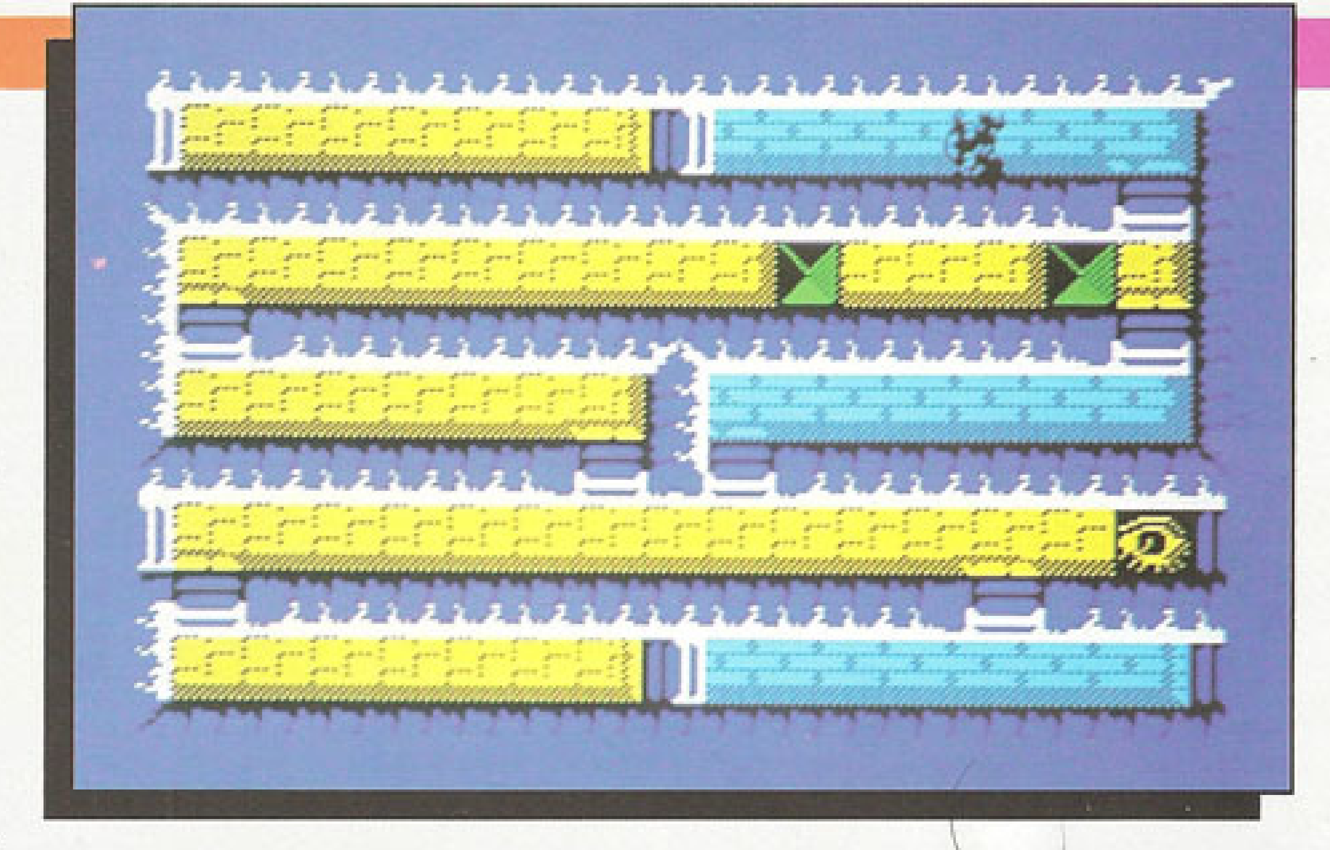
Sin embargo, el desdichado Mervyn pronto tuvo motivos para agradecerse a sí mismo su torpeza: los Warlocks, temibles magos nigromantes de los bosques del Norte, atacaron por sorpresa la misma noche de la transformación, y mientras los compañeros de Mervyn, alumnos de la escuela de «artes mágicas», morían o eran capturados a manos de los invasores, él, protegido por su inocente apariencia de rana minúscula, pasó desapercibido y pudo vivir para contarlo.



Desde entonces en su pequeño cerebro de rana sólo hay sitio para un único pensamiento obsesivo: vengar a los suyos y recuperar su apariencia humana... (¡Con tu ayuda, claro!)

El juego se estructura en la nada despreciable cifra de 64 niveles de 16 pantallas cada uno, repartidos en ocho grandes bloques denominados *mazmorras*. Como cada mazmorra es idéntica a las demás, la secuencia de mapeado se repite cada ocho niveles. En cada nivel hay doce Warlocks, entre magos y nigromantes (estos últimos, mucho más peligrosos). En los niveles inferiores, los nigromantes son bastante escasos, mientras que los magos abundan en una proporción de diez a uno sobre éstos. También existen cantidades inagotables de guardianes y armas mágicas, que se generan espontáneamente desde unos artefactos colocados a tal efecto en casi todas las habitaciones y pasillos. En menor número, se encuentran grabados en el suelo una serie de símbolos mágicos, llamados GLIFOS, que constituyen la única ayuda de que disponemos. Todos ellos sirven para protegernos de nuestros enemigos, pues son lugares sagrados que éstos no se atreverán a pisar. Aparte de los poderes temporales que confieren los GLIFOS (p. ej., ver un mapa en que aparecen las salas exploradas), siempre llevaremos con nosotros cuatro hechizos distintos, cada uno de naturaleza diferente, que son la clave del juego.

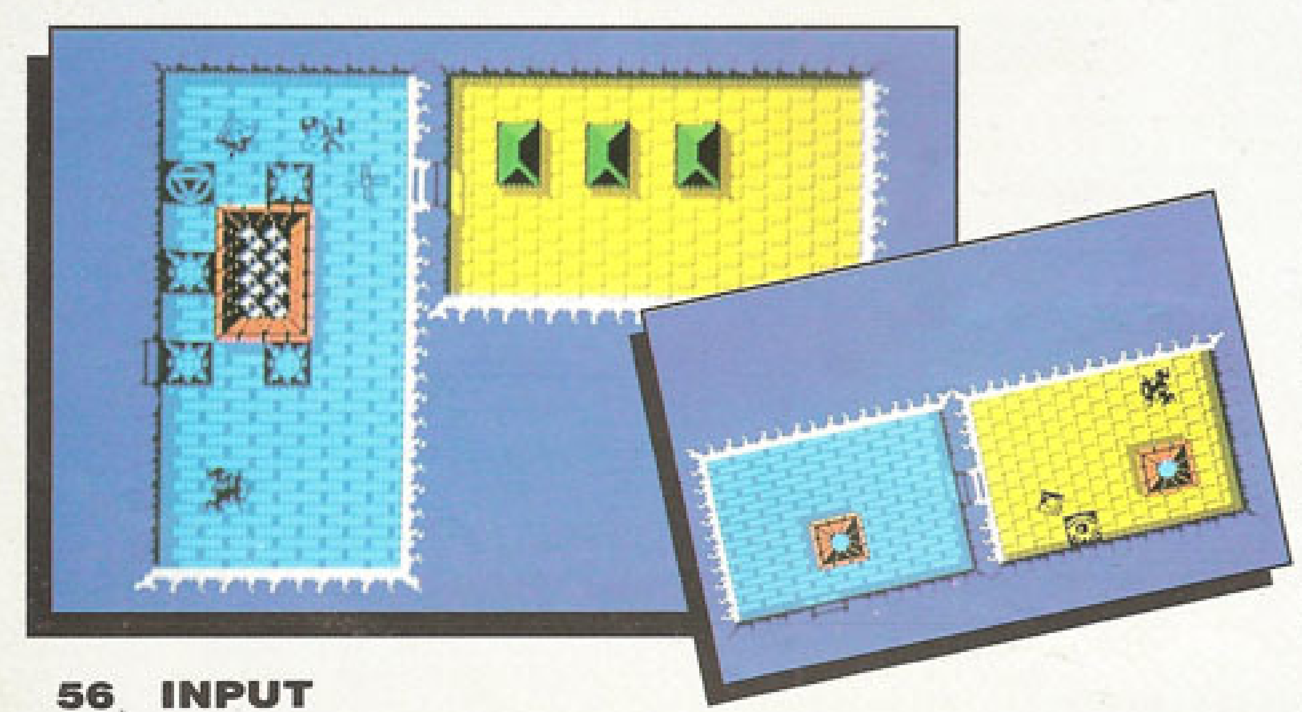
LOS HECHIZOS



Existen 32 hechizos distintos, agrupados en cuatro tipos de ocho niveles.
El primer tipo es el llamado POWER (poder), y sirve para proporcionar la energía que permita usar con éxito los demás hechizos, además de dar fuerzas por encima del nivel natural de resistencia.
El segundo tipo es el denominado DEFENSA, y como su nombre indica, su única utilidad es la de protegernos en el combate físico contra nuestros enemigos.
El tercero, llamado ATAQUE, es el que determina el poder de disparo.
Por último, tenemos un cuarto tipo de hechizo, denominado «DE EFECTO», que permite disfrutar de una serie de poderes diversos, según su nivel.
Tanto la capacidad de disparo como la de defensa dependen de la relación que se establece entre el nivel mágico de nuestros hechizos, del 1 al 8, y el nivel de poder de cada enemigo, catalogado en la misma escala. Por ello, el principal objetivo del juego es obtener hechizos cada vez más poderosos, es decir, de un nivel más próximo al octavo, que nos permitan enfrentarnos con posibilidades de éxito a los guardianes más peligrosos. Entre tus adversarios más temibles debemos citar al Caballero Guardián (que es muy resistente y cuenta con poderes mágicos), la mortífera Serpiente, el Fantasma (que es casi invisible y absorbe una gran cantidad de energía), la Araña (que es muy peligrosa y persistente) y la Gárgola (que es el enemigo más poderoso y temible).

COMBATE RITUAL

Para conseguir los hechizos, es necesario reunir determinadas combinaciones de RUNAS, que los Warlock guardan celosamente. Cada vez que nos encontramos con un mago o un nigromante, debemos entablar un «combate ritual» que nos permitirá derrotarlos de forma que sus conocimientos, transmitidos en forma de RUNAS, pasen a nosotros.
El «combate ritual» tiene lugar con sólo tocar al Warlock. Cuando esto ocurre, aparece en la pantalla la palabra RANARAMA con sus letras desordenadas, y se nos da un tiempo límite variable (según el poder del Warlock y la eficacia de nuestros hechizos) para recomponerla. Si lo logramos, volvemos al escenario de la acción y nos encontramos con unos destellos, las RUNAS, que debemos capturar antes de que se desvanezcan para siempre. Después, si queremos activar con ellas nuevos hechizos, debemos acudir al GLIFO de MAGIA más próximo, y operar en él con las pantallas de información-activación correspondientes.
La derrota en un «combate ritual» supone la pérdida del hechizo de poder que se posea. En su lugar, se obtiene otro hechizo de nivel mínimo (nivel MORTAL). Cuando, encontrándose en este nivel mínimo, se sufre una derrota (ya sea en combate ritual o en combate físico), o se agota toda la energía por desgaste, entonces se acaba el juego.

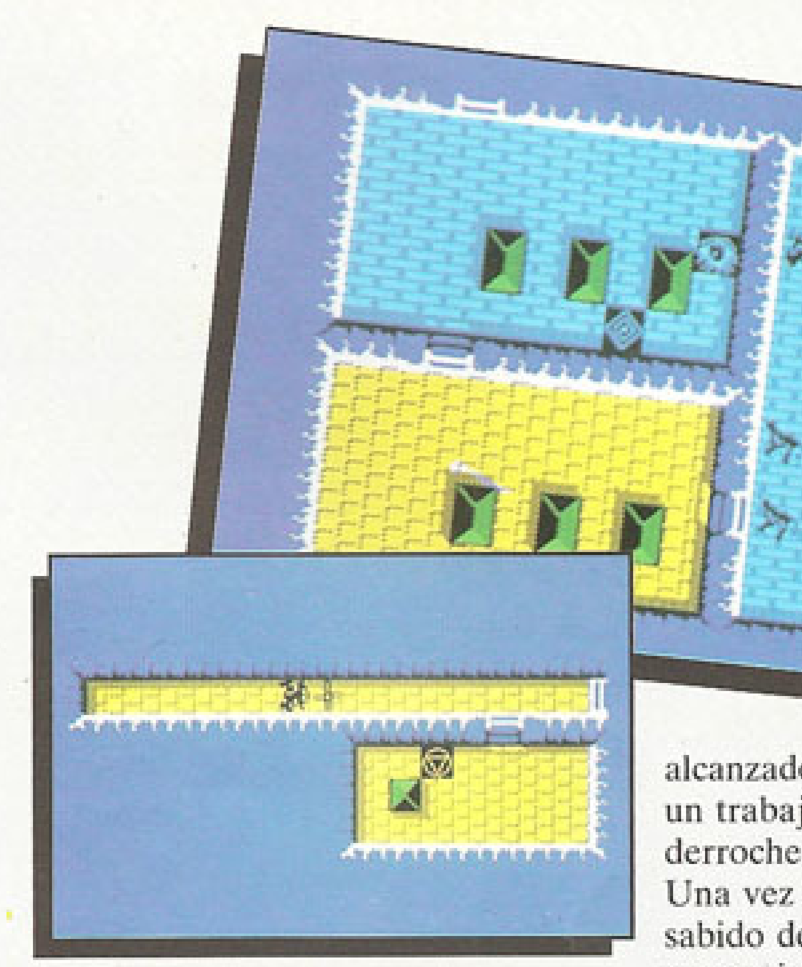


CONSEJOS

Debes procurar que los hechizos de poder tengan siempre un nivel más alto que los de defensa, ataque y efecto, ya que, de lo contrario, tu energía se agotará rápidamente.
No sobrevalores los hechizos de efecto: el segundo (SEE), a pesar de su bajo nivel, puede resultar más útil que otros teóricamente más poderosos.
Nunca accedas a un nivel superior en cada mazmorra sin antes asegurarte de que tus hechizos son iguales o

mayores al nivel de poder de los enemigos que te vas a encontrar. Por ejemplo, sería temerario pasar al segundo nivel sin haber conseguido hechizos de nivel superior a uno. Durante el combate físico, los GLIFOS pueden ser de una ayuda inestimable. Si te colocas sobre alguno de ellos, no sólo estarás protegido del roce mortal de los guardianes y las armas mágicas, sino que además podrás disparar contra ellos como si de un «tiro al blanco» se tratase. No obstante, no es conveniente entretenerse demasiado matando enemigos secundarios, ya que supone perder tiempo y energía sin recibir ningún beneficio apreciable a cambio. Si en alguna estancia la presencia de enemigos se hace agobiante, la mejor táctica consiste en destruir los generadores que los producen.

Hacer un mapa no es una estrategia tan útil como pueda parecer al principio: los GLIFOS no aparecen siempre en los mismos lugares, y los puntos de transporte mágico abundan lo suficiente como para encontrarlos rápidamente con sólo buscar al azar en cualquier dirección. Por otra parte, algunos GLIFOS permiten ver un plano general en el que se señala el punto donde nos encontramos, y las habitaciones que hemos recorrido hasta el momento. Por último, no debemos confundir los «cristales de energía» que aparecerán flotando en muchas estancias, con enemigos que vienen a robarnos fuerzas. Los cristales de energía, reconocibles por su forma de diamante, refuerzan el hechizo de poder que portamos, retardando su agotamiento. Cuando la rana empiece a parpadear, será señal de que su energía se acaba, y, por tanto, para evitar una muerte segura, no hay más solución que activar un nuevo hechizo de poder (si tenemos runas suficientes para ello), o buscar los preciados cristales.



VALORACION

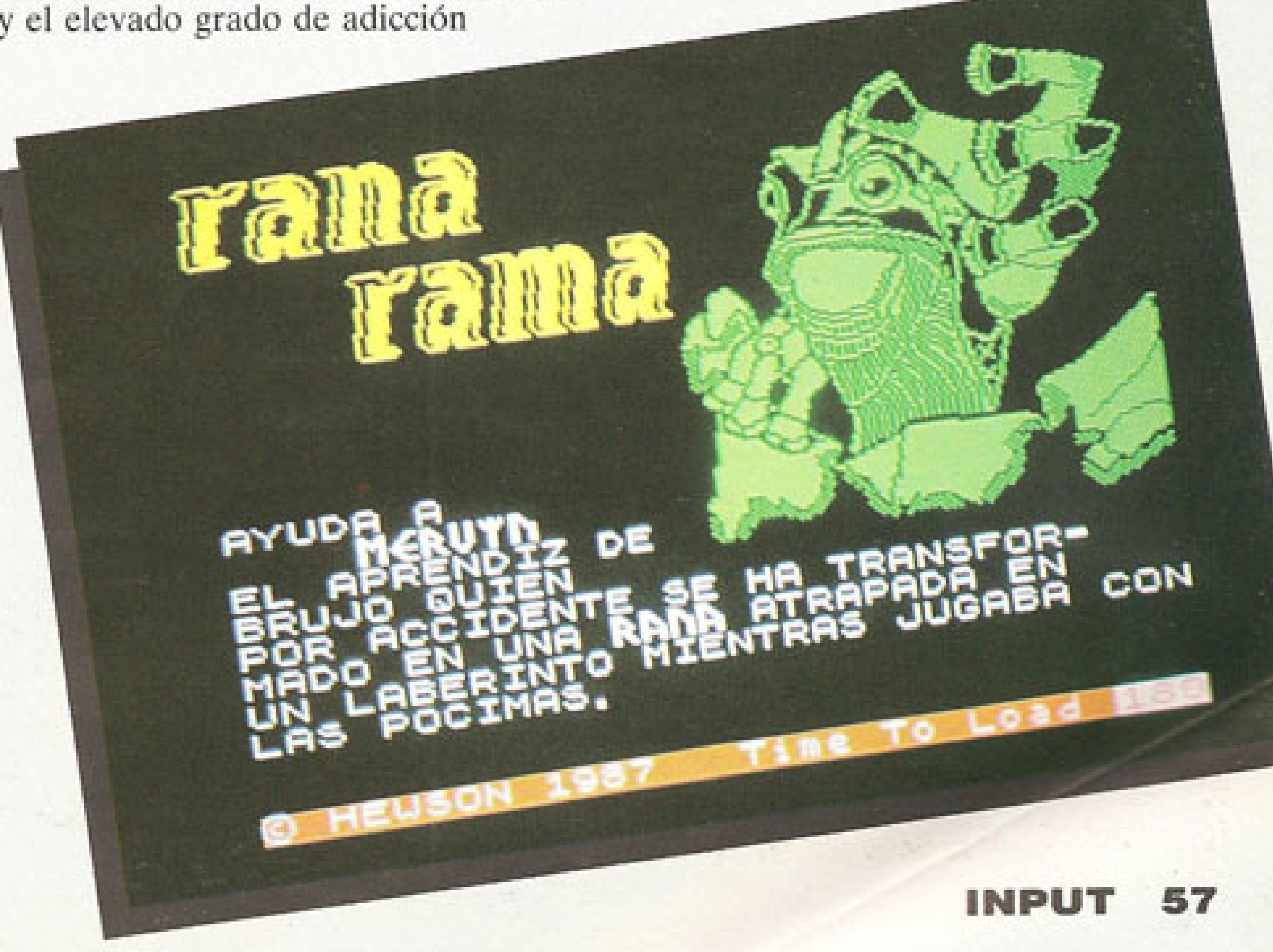
RANARAMA es, sin ninguna duda, la más adictiva vídeo-aventura de la temporada, y probablemente uno de los mejores juegos que han visto la luz en los últimos tiempos.

Como no hemos logrado encontrar ni el más mínimo detalle negativo, y tampoco estaría bien que nos deshiciéramos en excesivos elogios, más vale que nos remitamos a la tabla de calificaciones y os dejemos juzgar a vosotros.

Terminaremos, eso sí, señalando la extraordinaria calidad de los gráficos y el elevado grado de adicción alcanzado, detalles que evidencian un trabajo minucioso y un enorme derroche de imaginación.

Una vez más, la firma HEWSON ha sabido demostrar que lo más comercial no tiene por qué ser a la vez lo menos original. Ojalá cunda el ejemplo.

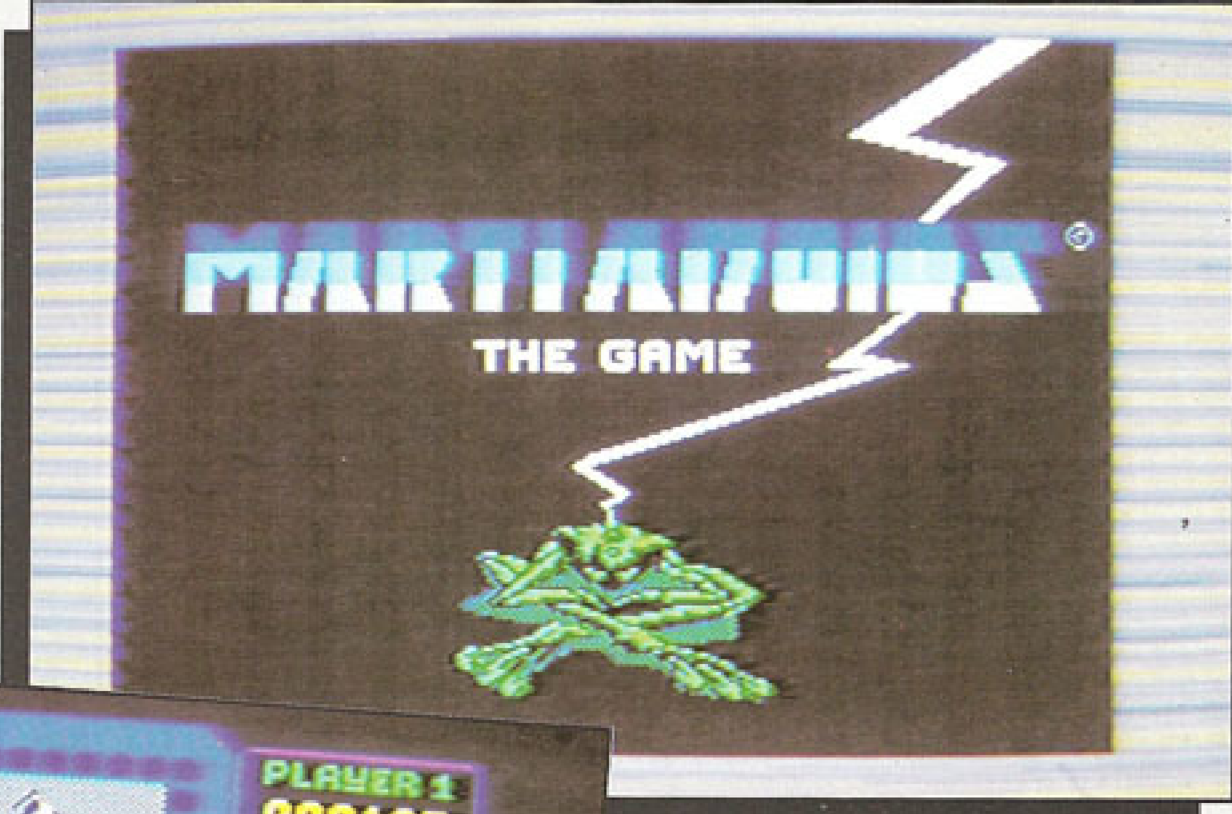
ANIMACION	9
INTERES	10
GRAFICOS	10
COLOR	10
SONIDO	8
TOTAL	47



MARTIANOIDS

● ULTIMATE ■ VIDEO-AVENTURA GRAFICA

Después de un largo viaje, la nave autotripulada MARKON I ha sufrido un ataque masivo de los bárbaros alienígenas. La mayor parte de los componentes activos han sido dañados, los programas se han borrado de la memoria de datos, y los robots de defensa han sucumbido en el combate. Sólo un milagro podría impedir que los invasores







MARTIANOIDS alcanzaran su propósito: capturar la nave para copiar su moderna tecnología y construir una flota capaz de invadir la Tierra.

Tu misión consiste en reparar y proteger al supercomputador MARKON, destruyendo a los invasores, e impedir que éstos consigan su objetivo.

Para ayudarte un poco, tu consola te informará en todo momento de tu posición en la nave y la situación de los sectores que debes defender.

Tanto en Gran Bretaña como en España este programa está provocando gran división de opiniones. Mientras unos dicen que no está en la línea habitual de ULTIMATE, sino mucho más abajo, otros afirman lo contrario. En nuestra opinión, MARTIANOIDS merece una calificación intermedia, teniendo en cuenta que presenta notables desequilibrios entre unos aspectos calificables como muy positivos, y otros más bien mediocres.

ANIMACION	9
INTERES	7
GRAFICOS	7
COLOR	6
SONIDO	7
TOTAL	36

TARZAN

● MARTECH ■ VIDEO-AVENTURA

La firma MARTECH ha publicado en España un nuevo programa que tiene por protagonista nada menos que al mítico TARZAN, sin faltar en el reparto la bella JANE, ni los malvados WAMABO, presentes en casi todas las películas del héroe de la selva.

Asi pues, el Rey de la Selva, luego de pasearse por las pantallas cinematográficas de toda España, encarnado por el actor francés Christopher Lambert en la película **Greystoke**, de Hugh Hudson, llega a las pantallas electrónicas con su bagaje de emocionantes aventuras por la misteriosa jungla, todavía virgen y no hollada por el imparable avance de la civilización.

Én esta ocasión, TARZAN se enfrenta a una peligrosa vídeo-aventura digna del propio Johnny Weissmuller —el recordado Tarzán cinematográfico—, con un argumento clásico del género: las siete piedras preciosas conocidas como «los ojos del arco iris» han sido robadas del templo de los Wamabo, y éstos, temiendo una terrible venganza de sus coléricos dioses, han raptado a JANE para ofrecerla en sacrificio.

Además de combatir contra los guerreros de la tribu Wamabo, nuestro fornido héroe deberá enfrentarse a todo tipo de fieras, y superar numerosas trampas y obstáculos si quiere volver a ver con vida a su JANE.

Los creadores del programa han ideado un original sistema que permite recrear el ambiente selvático con bastante realismo, colocando unas sombras de árboles en primer término, que dan la sensación de que el espectador contempla la escena oculto entre la maleza, como si estuviera dentro del escenario. Pero esto, unido a un desarrollo gráfico profuso en pantallas bicromáticas, dificulta a veces la visión de lo que está ocurriendo, sobre todo si repentinamente se hace de noche, y obliga a mantener un elevado grado de concentración en el juego, provocando un rápido cansancio.



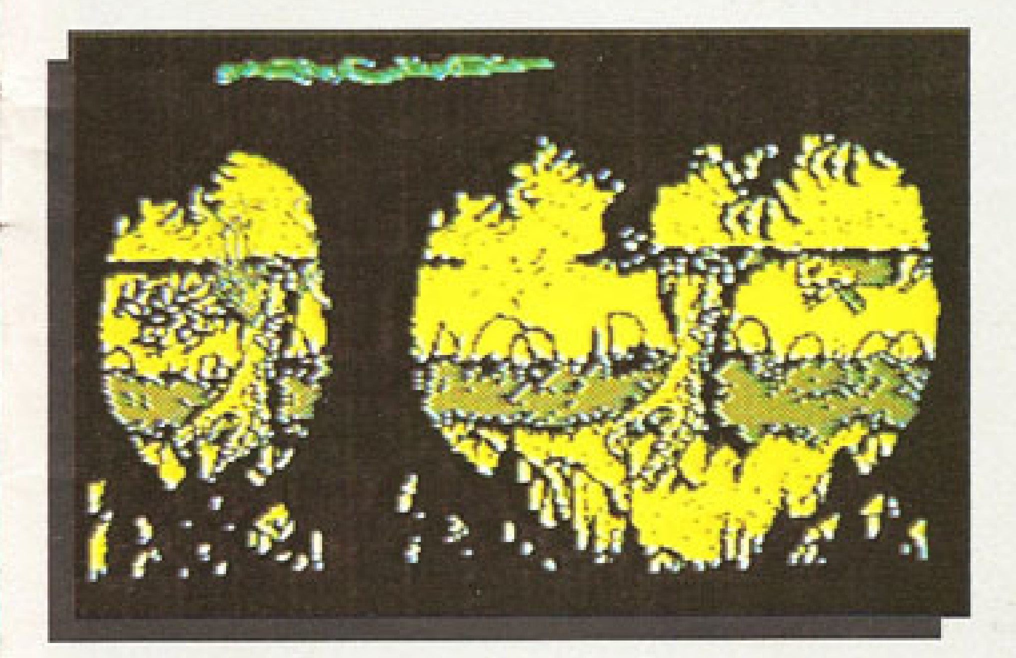
Por otra parte, el nivel de adicción es bastante bajo, a pesar de que en líneas generales se trata de un programa de gran calidad, con detalles (como el diseño gráfico), muy acertados.

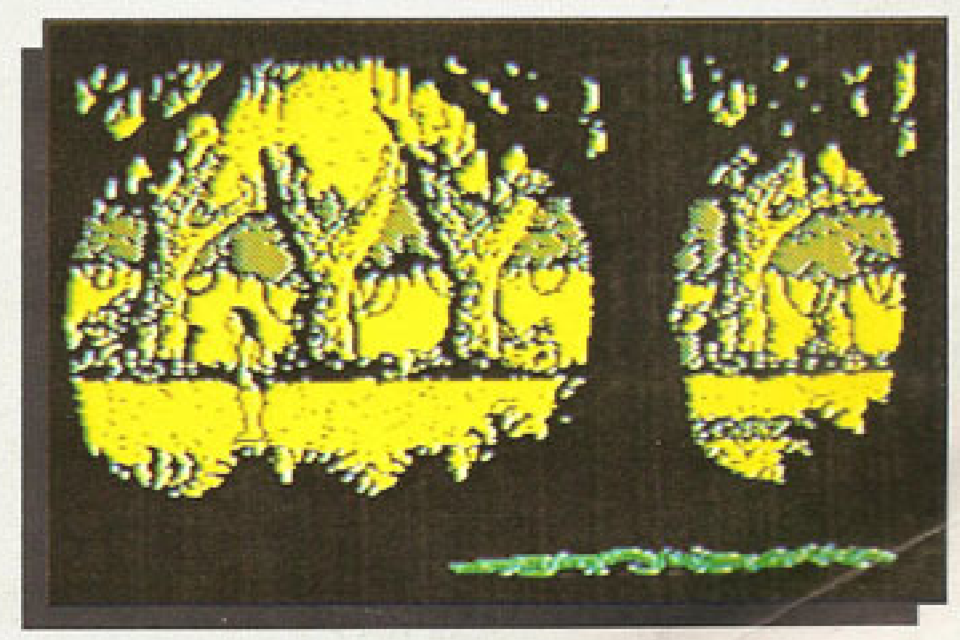
Antes de terminar, os haremos algunas recomendaciones:

En varias pantallas, aparecerán unos cofres negros disimulados sobre el fondo de la escena, difíciles de identificar, en los que se pueden encontrar objetos útiles para completar la aventura, como linternas, sogas y escudos. También existen numerosas zanjas y diversos obstáculos igualmente ocultos, por lo que es conveniente avanzar siempre con sumo cuidado, economizando al máximo las energías.

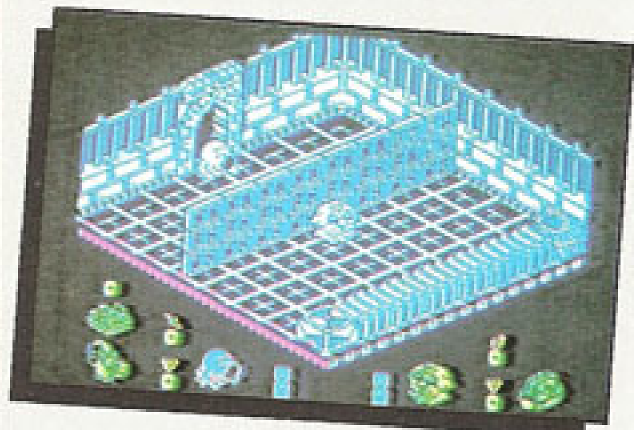
En cuanto a los enemigo persistentes, lo más práctico es usar el «salto acrobático» (propio del mejor programa de artes marciales) para escapar, sin perder el tiempo en combates.

ANIMACION	7
INTERES	6
GRAFICOS	8
COLOR	7
SONIDO	4
TOTAL	32





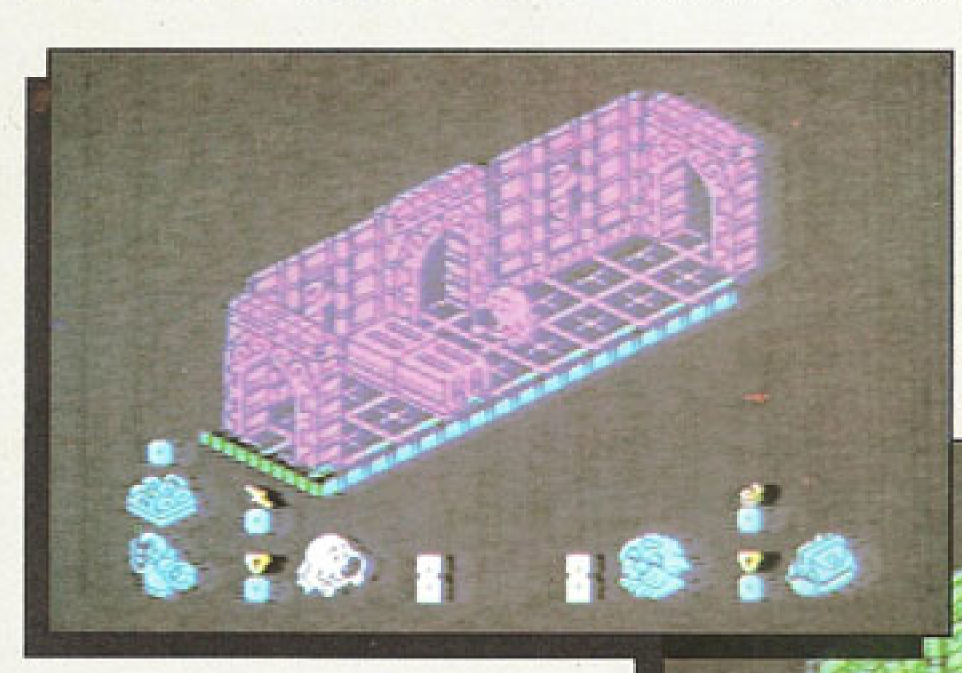




HEAD OVER HEELS

● OCEAN ■ VIDEO-AVENTURA

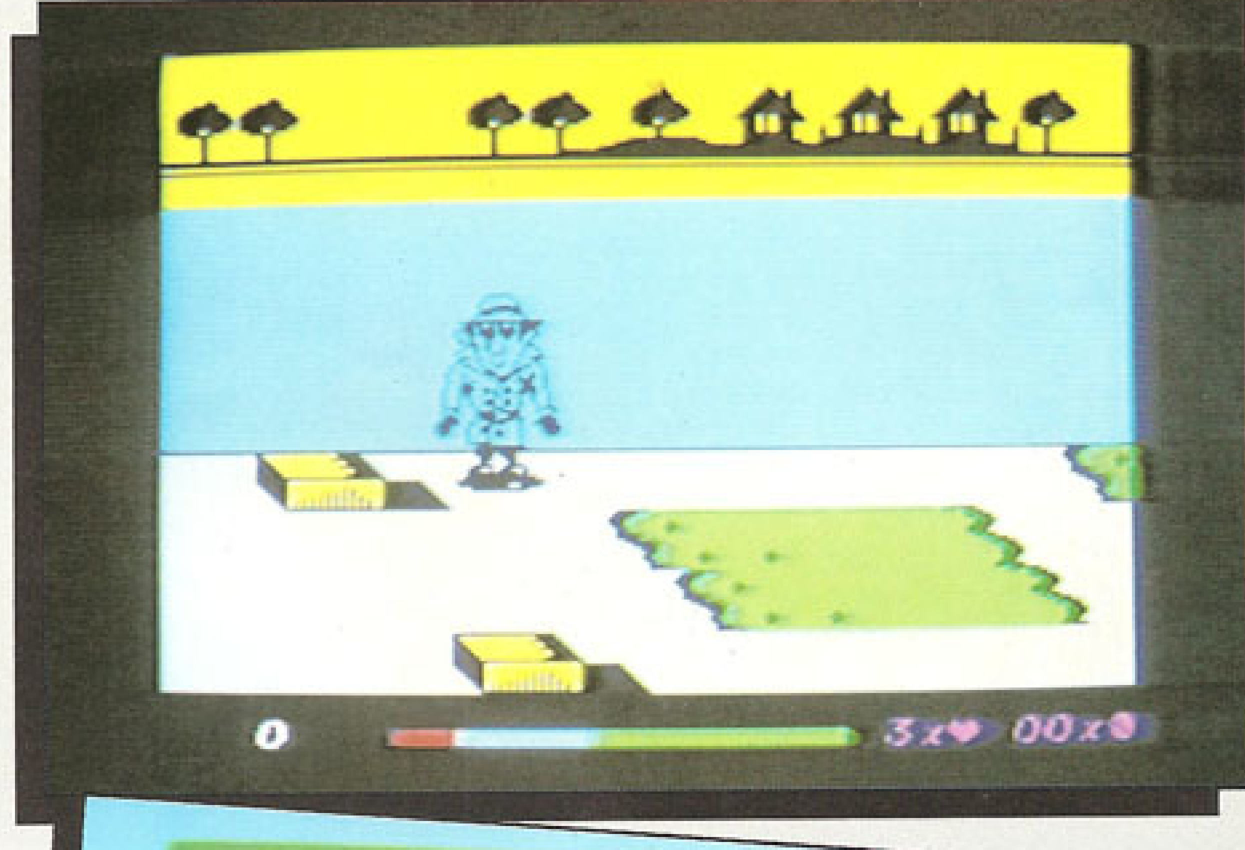
Recogiendo el modelo creado por ULTIMATE con el famoso KNIGHT LORE, OCEAN ha publicado un nuevo FILMATION en el que se unen todos los elementos que, por separado, dieron el éxito a muchos otros programas, como la perspectiva tridimensional en proyección isométrica, o la coexistencia de dos protagonistas que trabajan en equipo.
El argumento se resume en las peripecias de dos curiosos seres simbióticos, llamados HEAD (cabeza) y HEELS (talones), capaces de unirse en un solo cuerpo y de separarse después para actuar cada uno por su cuenta.
HEAD puede dar grandes saltos y disparar *donuts* con su trompa mágica. HEELS, en cambio, apenas da pequeños saltitos y no sabe disparar, pero puede transportar objetos en su bolso.
La misión que aguarda a ambos personajes consiste en liberar de la tiranía a los cinco planetas controlados por el imperio de BLACKTOOTH, encontrando las cinco coronas que simbolizan su libertad. Para lograrlo, deberán aprender a repartirse el trabajo según las aptitudes de cada uno, y poner en práctica todas sus habilidades. La pericia y los buenos reflejos jugarán siempre un papel fundamental.
Con un elevadísimo número de pantallas (más de 300) y un diseño gráfico realmente impresionante, el equipo de programadores que desarrolló al mítico BATMAN ha conseguido dotar a HEAD OVER HEELS de todo lo que se puede pedir a un buen juego. Esperamos que el próximo sea todavía mejor. ¡Bienvenidos a España estos dos simpáticos personajes.



ANIMACION	10
INTERES	9
GRAFICOS	10
COLOR	8
SONIDO	8
TOTAL	45

INSPECTOR GADGET

● MELBOURNE HOUSE ■ ARCADE



El famoso INSPECTOR GADGET vuelve a la carga en su lucha contra los agentes de MAD. De nuevo, deberá usar sus «Gadgetopoderes» para culminar con éxito otra arriesgada misión: desactivar las bombas que han sido colocadas en todos los circos de la ciudad.
Con un nivel de calidad técnica muy inferior al habitual en la mayoría de sus producciones, la firma MELBOURNE HOUSE acaba de publicar en España la versión «micro» de las aventuras de este peculiar «superdetective», respetando los rasgos caricaturescos de la serie de dibujos animados.
Con la «invisible» ayuda de su dulce sobrina y del heroico y transformista can que los acompaña, el susodicho inspector hará dar al traste las perversas maquinaciones de la cohorte de esbirros del doctor Gang. Las sobremesas televisivas dominicales tendrán, de esta manera, continuación en la pantalla de nuestro ordenador.
Él programa se desarrolla sin ningún tipo de complicaciones, con un nivel de dificultad quizás excesivamente bajo. Probablemente, los autores han querido conseguir de esta forma que el juego esté al alcance del público de la serie, mayoritariamente de muy corta edad, eliminando todo aquello que pudiera añadir alguna complejidad a la acción.
En suma, podemos decir que se trata de un programa muy indicado para los más pequeños de la casa, pero excesivamente simple y poco adictivo para los usuarios acostumbrados a aceptar grandes retos.



ANIMACION	6
INTERES	5
GRAFICOS	5
COLOR	4
SONIDO	4
TOTAL	24

TRES NUEVOS TITULOS DE FIREBIRD

Con un abultado catálogo de más de 25 títulos, y los más curiosos procedimientos de venta, la serie SILVER de FIREBIRD sigue acaparando una parte importante del mercado británico, junto a la omnipresente MASTERTRONIC. Sus expositores de venta pueden encontrarse en los más insólitos lugares, no sólo en tiendas especializadas y grandes almacenes, y sus precios (unas 500 pts.) no tienen competencia (no olvidemos que la operación «bajada de precio» sólo ha tenido lugar en España: en Gran Bretaña se sigue pagando por un «número uno» el equivalente a 2.000 pts.).

Pacientemente, la firma española DRO SOFT ha venido publicando en nuestro país la mayor parte de estos títulos, lanzando varios cada mes. Esta vez les ha tocado el turno a THRUST II, HELICHOPPER y WILLOW PATTERN.





THRUST II

THRUST II es la segunda parte de un interesante arcade que tuvo cierto éxito durante el año pasado, gracias a un planteamiento sencillo y adictivo a la vez.

Continúa la guerra con el Imperio. Las fuerzas rebeldes han capturado un pequeño planeta artificial, pero su atmósfera, aún sin purificar, impide que sea útil a los propósitos de sus ocupantes. La misión consiste en extraer del fondo del planeta los depósitos de componentes químicos que pueden limpiar definitivamente la atmósfera, y acabar con los pocos guardianes del Imperio que aún quedan en el interior.

Un arcade bien hecho, y sin duda uno de los mejores programas de FIREBIRD.

HELICHOPPER

Se trata de un programa de rescate con 23 pantallas, en el que deberás poner a salvo al mayor número posible de CLONES, maniobrando entre diversos obstáculos y enemigos. Desde el punto de vista gráfico, el programa deja mucho que desear, pero al menos no le falta una buena dosis de adicción.

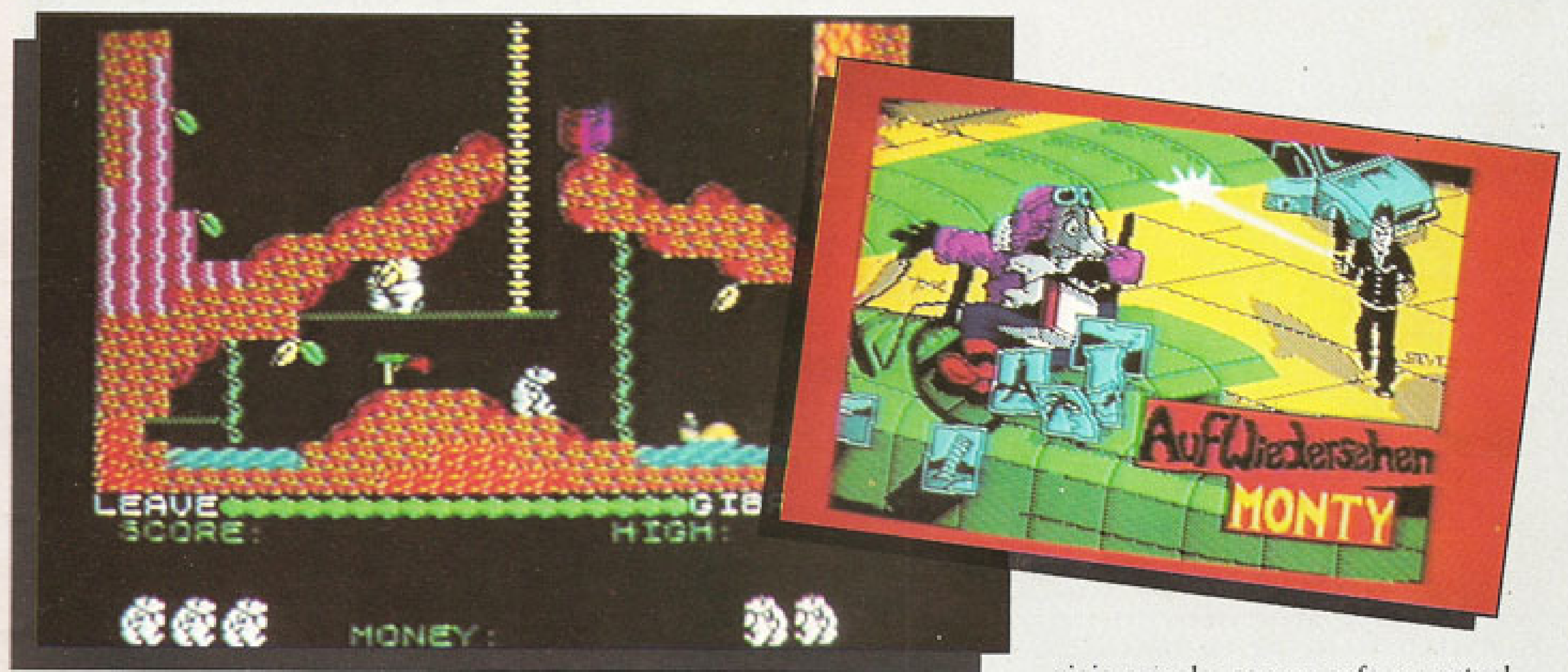
WILLOW PATTERN

La aventura transcurre en el interior de un laberíntico jardín oriental, plagado de samurais, donde se encuentra encerrada una princesa. Tu misión, naturalmente, consiste en rescatarla. Para lograrlo, deberás derrotar los samurais que te corten el paso, y usar adecuadamente varios objetos. Existen ciertos lugares en los que se produce un espectacular cambio de ambientación, que consigue dotar al juego de una especial originalidad, sobre todo si tenemos en cuenta que no sólo cambia el escenario, sino también el planteamiento básico del juego: se trata de pruebas aisladas de habilidad y reflejos, casi al margen del desarrollo de la aventura.

En definitiva, un programa original con buenos gráficos y la dosis justa de acción.







AUF WIEDERSEHEN, MONTY

● **GREMLIN** ■ **ARCADE**

AUF WIEDERSEHEN, MONTY es una nueva aventura del popular MONTY MOLE, en la que se repite una vez más el tradicional modelo de arcade tipo MANIC-MINER usado en las aventuras precedentes.
El juego está ambientado en 10 países europeos, incluido España, a través de los cuales Monty deberá buscar el dinero que necesita para comprar la isla de Montos, un lugar paradisíaco donde piensa retirarse a vivir.

El nivel de dificultad no es tan alto como el habitual en la mayoría de los juegos de su clase, pero esto, lejos de restar interés al programa, contribuye a hacerlo tremendamente adictivo desde la primera partida. Por otro lado, la gran variedad de gráficos, situaciones y personajes, hacen de cada pantalla un microcosmos con personalidad propia, hasta el punto de que en ocasiones hay escenas tan distintas que parecen corresponder a programas diferentes.
En definitiva, un buen arcade de la vieja escuela, con un enfoque actual. Un detalle anecdótico: cuando Monty llega a un país, se oye el himno nacional de éste. Sin embargo, cuando entra a España se oye *Que viva España*, de Manolo Escobar.

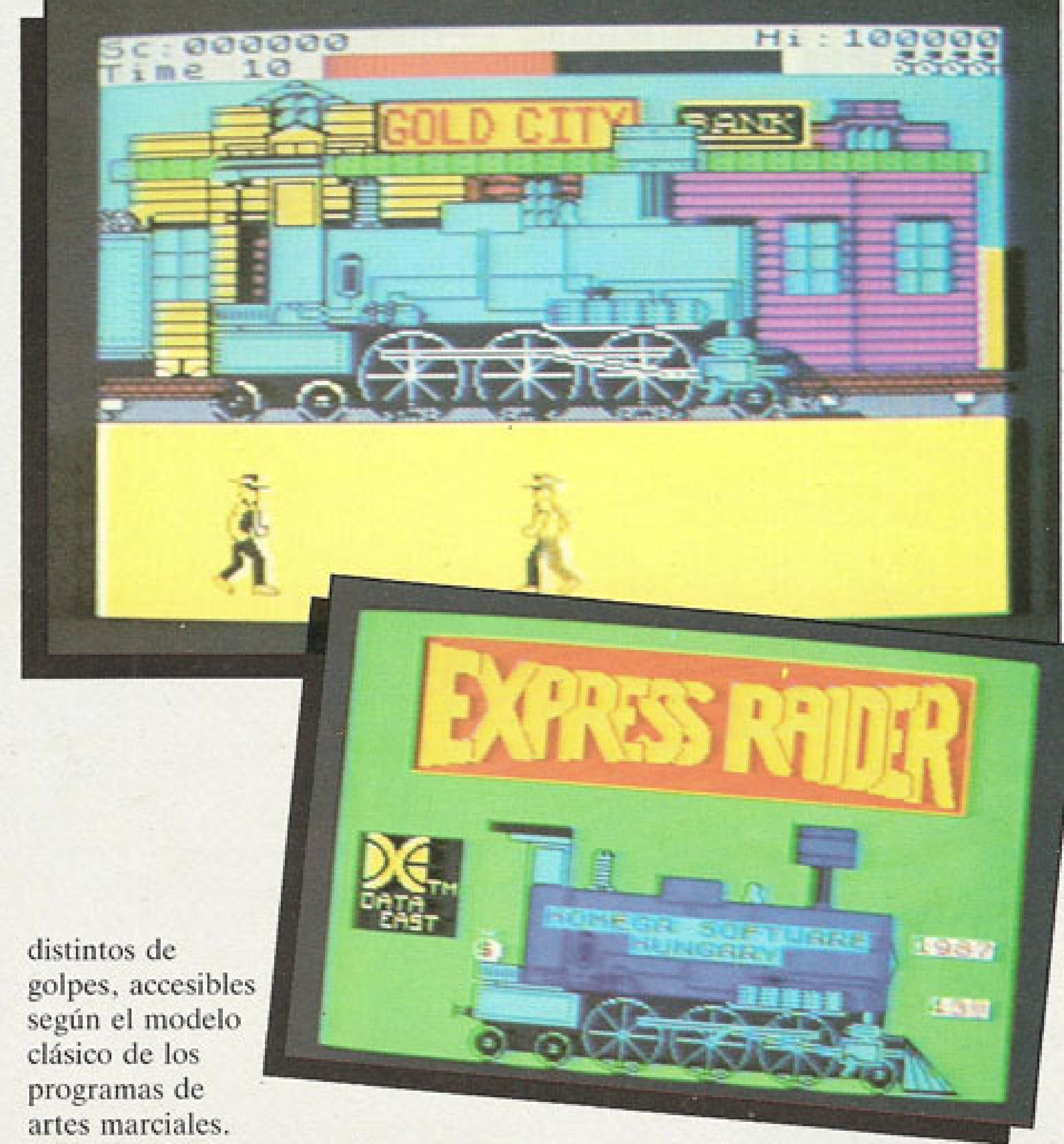
ANIMACION	8
INTERES	9
GRAFICOS	9
COLOR	7
SONIDO	7
TOTAL	40



EXPRESS RAIDER

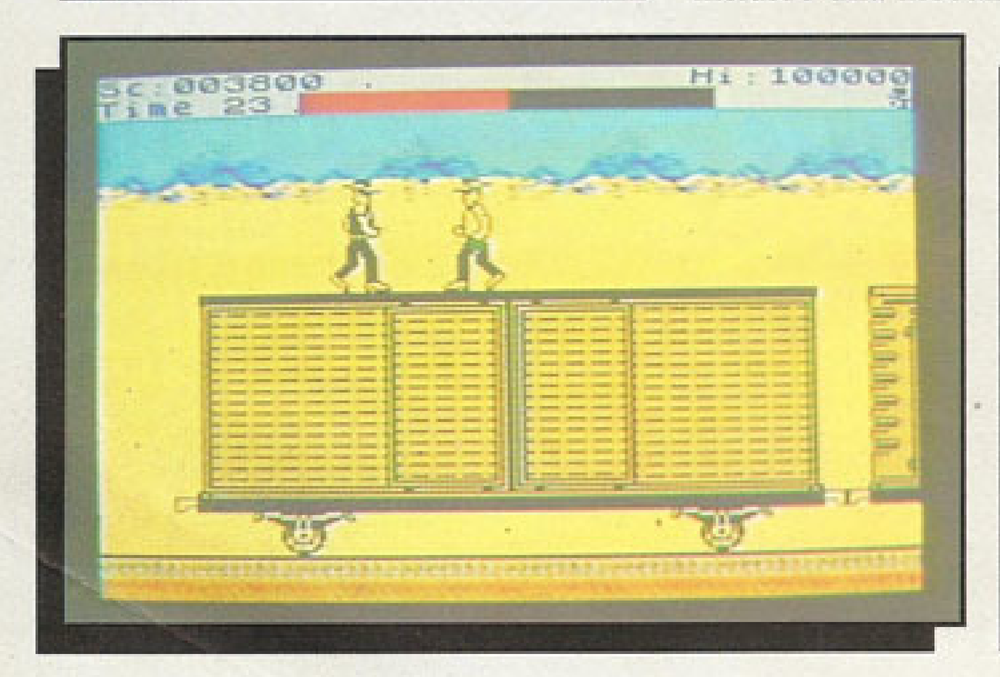
● **US GOLD** ■ **JUEGO**

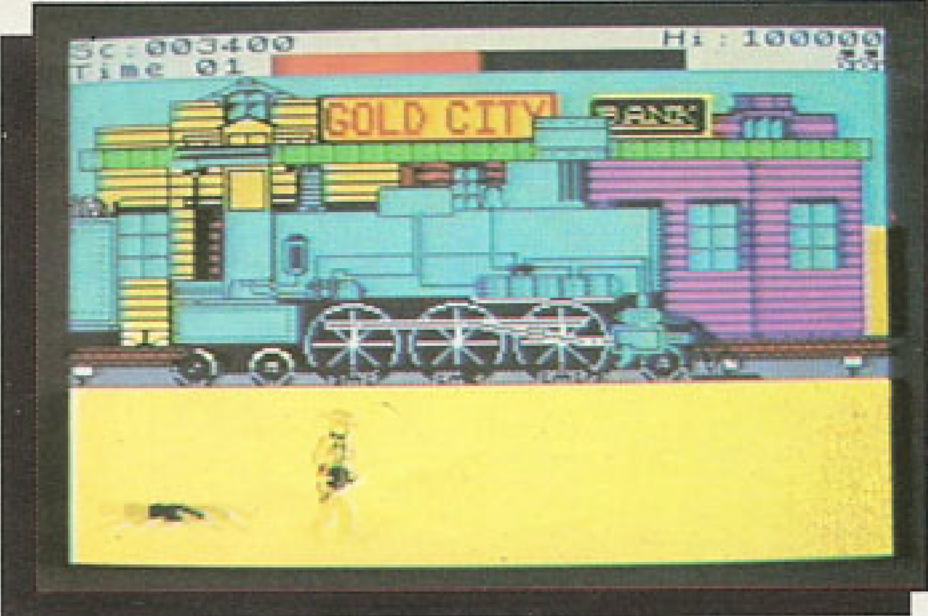
¡POR FIN!
Después de haber figurado durante meses en las páginas de publicidad de todas las revistas de software, por fin ha sido publicada en España la versión SPECTRUM de EXPRESS RAIDER, un magnífico programa que nos traslada a los tiempos del lejano Oeste, cuando asaltar trenes y pelear sobre vagones en marcha era algo habitual.
Deberás hacerte con el control de un tren andando por el techo de los vagones. Verás cómo, al instante, los techos se llenan de gente con cara de pocos amigos que te atizarán a la primera de cambio. Pero los peligros también vienen de los aires: numerosas aves pueden abatirse, sobre ti causándote daños que harán bajar tu indicador de fuerza.
Desarmado, deberás derrotar a los enemigos combatiendo en fases sucesivas de gran dificultad, usando tan sólo tu habilidad con los puños y tus reflejos en el asalto. Dispones de 8 movimientos diferentes y 6 tipos distintos de golpes, accesibles según el modelo clásico de los programas de artes marciales.



ANIMACION	8
INTERES	8
GRAFICOS	9
COLOR	8
SONIDO	5
TOTAL	38

Un detalle interesante es la posibilidad de seleccionar una opción que hará inútiles los esfuerzos de los buscadores de pokes, y que seguramente entusiasmará a los menos expertos: para no desesperar a los principiantes, los autores han incluido una modalidad de juego con nada menos que ¡32 vidas!
Aparte de este interesante aspecto, también debemos valorar positivamente el nivel de adicción, el acertado diseño gráfico y, sobre todo, la diversidad de pruebas y escenarios.





SABOTEUR II

● **DURELL** ■ **VIDEO-AVENTURA**

SABOTEUR ha sido y sigue siendo uno de los mayores mitos de la historia del software. Está considerado como un ejemplo paradigmático de lo que debe ser un buen juego de acción, y su molde, desgraciadamente roto, ha sido imitado en repetidas ocasiones por muchos otros programas de éxito. Ahora, por fin podremos disfrutar de la segunda parte de esta sensacional videoaventura, publicada bajo el prestigioso sello de DURELL SOFTWARE.

Al igual que en la primera parte, el objetivo del juego consiste en «forrarse» de millones, cobrando fuertes sumas por realizar una serie de sabotajes en instalaciones enemigas. Concretamente, se trata de reunir todas las cintas perforadas que se encuentran dispersas en el Centro de Mando del Dictador, usándolas después para reprogramar los ordenadores de control, y desviar así la trayectoria de los misiles que van a ser lanzados de un momento a otro. Si además se consigue escapar con vida, comienza una nueva misión con mayor grado de dificultad.

La única vía de escape es un largo túnel situado en la zona más profunda de la base, al lado izquierdo de la pantalla, en el cual espera una moto que permitirá alejarse rápidamente. En la parte superior izquierda, están almacenados los misiles, listos para ser disparados, y en el lado opuesto se encuentra el arsenal. El resto de las instalaciones están ocupadas por pasillos subterráneos y zonas en construcción.

Hay dos tipos de enemigos en el interior de la base: los guardianes androides, de mayor poder y envergadura física, y las panteras negras, peligrosos animales salvajes que andan sueltos en algunos sectores. Ambos son muy resistentes a las precarias armas de que se dispone para completar la misión, y por ello resultan especialmente difíciles de derrotar. Cuando las circunstancias se pongan difíciles en el combate, lo mejor es huir rápidamente, sobre todo si hay varios enemigos en la misma pantalla, ya que normalmente son necesarios más de tres golpes para derrotar a un solo contrario, y perder una sola vida supone tener que empezar de nuevo. También se debe tener cuidado de no tocar el muro electrificado que rodea las instalaciones, pues su descarga produce la muerte instantánea.

La táctica más eficaz consiste en arriesgarse sólo por las acciones que resultan más rentables: terminar con éxito una misión, por ejemplo.

ANIMACION	8
INTERES	9
GRAFICOS	7
COLOR	7
SONIDO	8
TOTAL	39





EL ZOCO



Vendo Modems de Spectrum caseros para pasar programas por teléfono. Pablo Martínez. C/ Rodas, 19. 03180. Torrevieja, Alicante. Tl. (965) 71 01 86. Llamar a partir de las 4 de la tarde.

Vendo cassette-recorder especial para ordenador con manual, modelo Philips D-6260/30 por 5.500 pts, está valorada en 11.000 pts (con un mes de uso). Jacobo. Tl. (965) 22 25 38.

Deseo crear un CLUB de usuarios del ZX Spectrum y del MSX a nivel nacional e internacional. Escribir a: Roberto Ramos Pérez. P.° de los Fueros, 7, 1.° C. Sangüesa 31400. Navarra.

Intercambio/vendo juegos y otros programas de aplicaciones para el Spectrum 48K. Facilito lista con todos los títulos a cambio de los que se me ofrezca. Seriedad en la calidad de cintas y grabaciones. Angel María González García. Avd. Virgen de Argeme; 5, 1.° C. 10800 Coria. Cáceres. Tl. (927) 50 04 10.

Intercambio/vendo preferentemente juegos a nivel nacional e internacional. Poseo las últimas novedades. Roberto Ramos Pérez. P.° de los Fueros, 7, 1.° C. Sangüesa 31400. Navarra.

Vendo Spectrum, Interface Kempston y Programable, joystick Quickshot II, manuales en inglés y castellano, más un libro sobre el Spectrum. Llamar a Daniel Costa Royo. Tl: 308 25 59. Barcelona.

Vendo ordenador Dragon 200, un monitor color, 15 juegos, precio a convenir. Mercedes. Tl. (93) 230 40 57.

Intercambio/compro/vendo programas para el Spectrum Plus, además compro MODEM para dicho ordenador. Mandar ofertas a: Antonio Toribio Carreras. Polígono Puerta Madrid, sector Málaga, edf. Granada, puerta B, 3.° izq. 23740 Andújar. Jaén.

Cambio juegos Spectrum. También cambio y compro libros de código máquina. Enrique Alapont. C/ Maestro Valls 1-19. Valencia (46022). Tl: (96) 367 53 94.

Intercambio programas para el Spectrum 48K. Tengo muchas novedades (Bomb Jack 2, Ace of Aces, Short Circuit y muchos más). Javier Domínguez Tejero. C/ Ramón y Cajal, 22, 1.° D. Santurce. Vizcaya. Tl: (94) 461 80 33.

Vendo ZX Spectrum 48K, interface 1, Microdrive, 2 interfaces para Joystick, 2 joysticks de Spectravideo (Quick Shot I, II), 3 cartuchos de microdrive, 20 cassettes de juego, 10 cintas de cassette especial computador vírgenes. Todo nuevo. Sólo 25.000 pts. Pedro Jesús Arce Guerrero. Urbanización Las Brisas, 1. C/ Eolo. 6. Mairena de Aljarafe. Sevilla. 41927. Tl: 76 20 66.

Intercambio juegos, ideas y mapas del ZX Spectrum. Interesados escribir a: David Álvarez González. C/ Saavedra, 7 6.° E. 33208 Gijón. Asturias.

Club para todos los amigos aficionados al Spectrum y compatibles. Prometemos contestar a todas las cartas. I. Soft. Club. Bda/ Torresoto. C/ Triana, 4. 11401 Jerez de la Frontera. Cádiz. Tl: (956) 32 12 34.

Intercambio programas para el Spectrum Plus con usuarios de toda España, mandar lista a: Agustín Rodríguez Obel. Huelva, 62. Trigueros. Huelva.

Vendo o cambio por cassette o cualquier periférico (menos joystick) todas las revistas INPUT SINCLAIR aparecidas hasta la fecha. En caso de venderlas, por 5.000 ptas. También cambio programas de juegos y utilidades. Juan Gabriel Villena. C/ San Isidro, 27. Padul. Granada. Tl: (958) 79 02 64. Sólo tardes.

Vendo Hisoft Deupac ensamblador-desensamblador Gens-Mons, original y con instrucciones en castellano. Llamar a José Antonio. Lleida. Tl: (973) 76 05 48.

Compro Interface para Spectrum, así como joystick ambos en buen estado y precio asequible. También quisiera contactar con usuarios del Spectrum de alrededores de Navahermosa. L. Esteban Manzanares. C/ Prado, 21. 45150 Navahermosa. Toledo.

Vendo Spectrum Plus (del año 85) con fuente de alimentación, cables, manual de instrucciones (español), cinta de demostración (Goldstar en inglés), 3 juegos: The Drive In, Bubble Buster y el Underwurlde, con todas las instrucciones para llegar al final sin problemas (español). Por sólo 20.000 pts. Miguel García. Tl: (91) 719 24 46. A partir de las 15 h.

Vendo simulador de Spectrum y Simon's Basic (en cinta) para CBM-64 con manuales incluidos. Sólo 5.000 pts. Pedro J. Gómez Pérez. Avd. de la Raza, 31. 3.° A. 21002 Huelva. Tl: (955) 24 20 31.

Vendo Spectrum 48K, en buen estado (con cables, transformador, etc.) cassette marca COMPUTONE, interface Kempston y un joystick, cinta de demostración, programas de juegos, utilidades, aplicaciones, etc. Todo por 35.000 pts. A.L. Barros. G. Rubín, 7. Pontevedra 36001.

Cambio juegos de Spectrum; bélicos, de deportes y de aventuras por algunos de los siguientes: Terra Cresta, Jail Break, Infiltrator o por cargador universal de código máquina C/M. (Se estudiarán ofertas). L. Leonardo Lara Gómez. C/ Rosales Portón, 10. 1.° D. Ceuta.

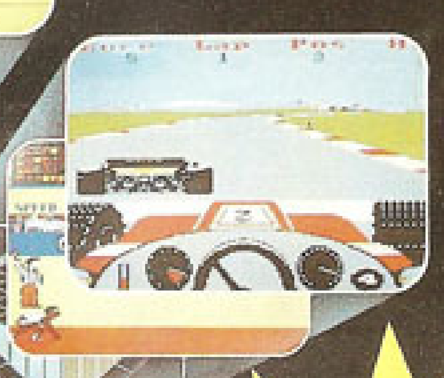
Vendo ZX Spectrum Plus seminuevo con todos sus accesorios; cables, transformador y además interface Kepston. Todos los manuales del ordenador y dos libros de informática aplicada. Cinta guía de funcionamiento y tres cintas de juegos. Todo por 26.000 pts. Adrián Sánchez Gómez. C/ Eustasio Amilibia, 4. 3.° B. San Sebastián 20011. Guipúzcoa. Tl: (943) 46 63 70.



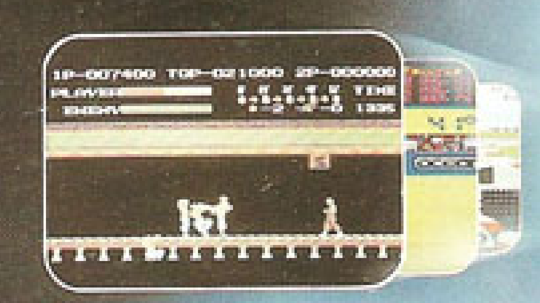
La maquina alucinante

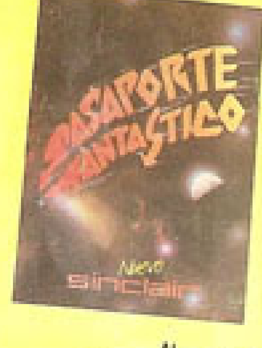






EL UNICO ORDENADOR CON MILES Y MILES DE PROGRAMAS DISPONIBLES.





Al comprar tu nuevo Spectrum pide el Pasaporte Fantástico. Podrás conseguir un reloj alucinante.

sinclair 128K ZX Spectrum +2

33.900 Pts.+ IVA

Microprocesador Z80A. 128 K RAM. 32 K ROM. Teclado de 58 teclas. 32 columnas × 24 filas de texto. Gráficos de alta resolución (256 × 192 pixels). 8 colores con dos niveles de brillo cada uno. Calculadora en pantalla. 3 canales de sonido programables e independientes. Cassette incorporada. Salida TV y monitor RGB.

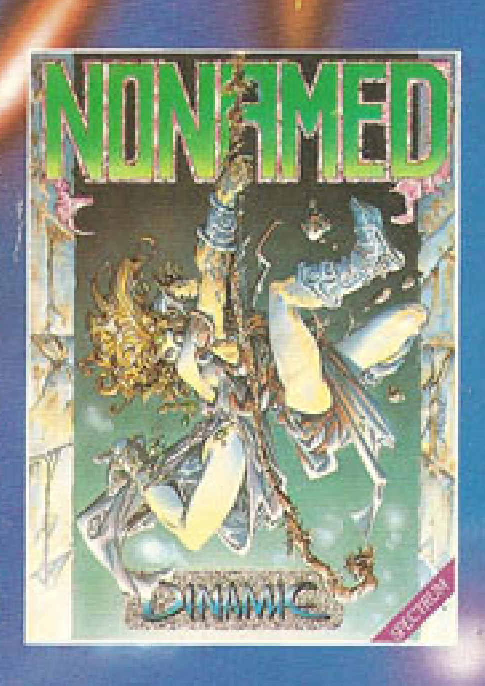
Interface MIDI (Musical Instrument Digital Interface). Salida Serie RS 232 bidireccional. Dos conectores para joysticks. Conector plano compatible con todos los modelós Spectrum anteriores. Editor de pantalla y dos versiones BASIC en ROM. 48 K BASIC, compatible con Spectrum 16 K, 48 K y ZX + . 128 K BASIC, compatible con ZX Spectrum 128.

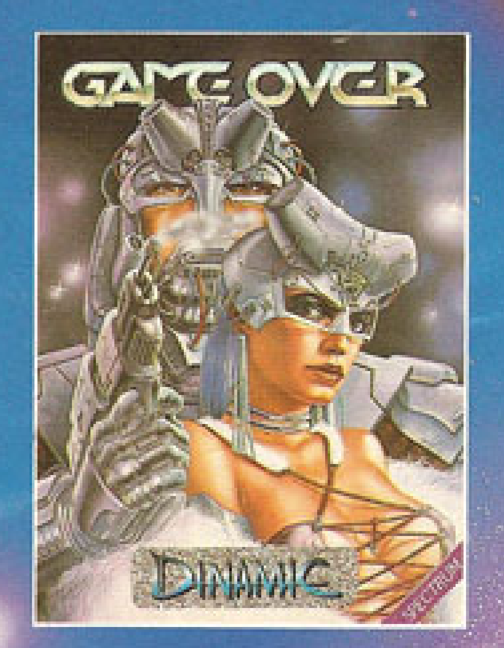
Nuevo sinclair ZX Spectrum +2

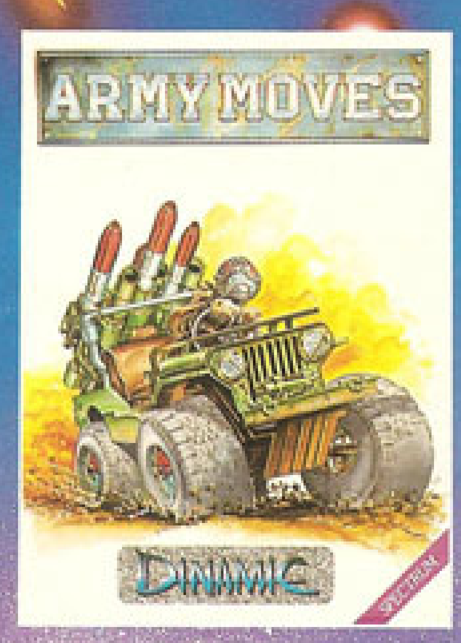
C/. Aravaca, 22. 28040 Madrid. Tel. 459 30 01. Telex 47660 INSC E. Fax 459 22 92. Delegación en Cataluña: C/. Tarragona, 110. Tel. 325 10 58. 08015 Barcelona.

ALTO VOLTAJE

NONAMED	GAME OVER	ARMY MOVES	DUSTIN
SPECTRUM • MSX AMSTRAD	SPECTRUM AMSTRAD	SPECTRUM • MSX AMSTRAD • CBM 64	SPECTRUM AMSTRAD

















★★★

875 PTS. CADA UNO, NUEVO PRECIO DINAMIC

★★★

¡¡INCREIBLE!!
LOS 4 JUEGOS EN UN DISCO AMSTRAD SOLO: 2.750 pts.

DINAMIC SOFTWARE. Plaza de España, 18. Torre de Madrid, 29-1. 28008 Madrid.
Pedidos contra reembolso (de lunes a viernes, de 10 a 2 y de 4 a 8 horas): Teléfono (91) 248 78 87.
Tiendas y Distribuidores: Teléfono (91) 447 34 10.

